



Red Hat OpenStack Platform 17.1

配置参考

配置 Red Hat OpenStack Platform 环境

Red Hat OpenStack Platform 17.1 配置参考

配置 Red Hat OpenStack Platform 环境

OpenStack Team
rhos-docs@redhat.com

法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

本文档适用于希望查找配置选项的系统管理员。它包含 OpenStack 可用的配置选项列表，并使用 auto-generation 生成各个项目的代码中的描述。

目录

前言	3
使开源包含更多	4
对红帽文档提供反馈	5
第 1 章 BARBICAN	6
1.1. BARBICAN.CONF	6
第 2 章 CINDER	31
2.1. CINDER.CONF	31
第 3 章 GLANCE	126
3.1. GLANCE-API.CONF	126
3.2. GLANCE-SCRUBBER.CONF	245
3.3. GLANCE-CACHE.CONF	301
第 4 章 HEAT	357
4.1. HEAT.CONF	357
第 5 章 IRONIC	396
5.1. IRONIC.CONF	396
第 6 章 IRONIC-INSPECTOR	470
6.1. INSPECTOR.CONF	470
第 7 章 KEYSTONE	492
7.1. KEYSTONE.CONF	492
第 8 章 NEUTRON	540
8.1. DHCP_AGENT.INI	540
8.2. L3_AGENT.INI	546
8.3. LINUXBRIDGE_AGENT.INI	553
8.4. METADATA_AGENT.INI	559
8.5. METERING_AGENT.INI	567
8.6. ML2_CONF.INI	572
8.7. NEUTRON.CONF	579
8.8. OPENVSWITCH_AGENT.INI	613
8.9. SRIOV_AGENT.INI	620
第 9 章 NOVA	626
9.1. NOVA.CONF	626

前言

本文档描述了 Red Hat OpenStack Platform 中每个主要服务的配置文件中可用的选项。内容会根据配置文件本身中的值自动生成，仅出于参考目的提供。



警告

不支持手动编辑配置文件。所有配置更改都必须通过 Director 进行。红帽仅作为技术参考提供本指南。

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [CTO Chris Wright 的信息](#)。

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。与我们分享您的成功秘诀。

在 JIRA 中提供文档反馈

使用 [Create Issue](#) 表单对文档提供反馈。JIRA 问题将在 Red Hat OpenStack Platform Jira 项目中创建，您可以在其中跟踪您的反馈进度。

1. 确保您已登录到 JIRA。如果您没有 JIRA 帐户，请创建一个帐户来提交反馈。
2. 点击以下链接打开 **Create Issue** 页面：[Create Issue](#)
3. 完成 **Summary** 和 **Description** 字段。在 **Description** 字段中，包含文档 URL、章节或章节号以及问题的详细描述。不要修改表单中的任何其他字段。
4. 点 **Create**。

第 1 章 BARBICAN

下面的章节包含有关 **barbican** 服务中的配置选项的信息。

1.1. BARBICAN.CONF

本节包含 `/etc/barbican/barbican.conf` 文件的选项。

1.1.1. DEFAULT

下表概述了 `/etc/barbican/barbican.conf` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
admin_role = admin	字符串值	用于将经过身份验证的用户标识为 administrator 的角色。
allow_anonymous_access = False	布尔值	允许未经身份验证的用户访问具有只读权限的 API。这只适用于使用 ContextMiddleware。
api_paste_config = api-paste.ini	字符串值	api 服务的 paste.deploy 配置的文件名
backdoor_port = None	字符串值	启用事件程序后端。可接受的值为 0、<port> 和 <start>:<end>，其中 0 会导致侦听随机的 tcp 端口号；<port> 会导致监听指定的端口号（如果正在使用该端口，则不会启用 backdoor）；<start>:<end> 会在指定的端口号范围内侦听最小未使用的端口号。所选端口显示在服务的日志文件中。
backdoor_socket = None	字符串值	启用 eventlet 后端，使用提供的路径作为可接收连接的 unix 套接字。这个选项与 <code>backdoor_port</code> 相互排斥，该选项只应提供一个。如果提供这两个选项，则此选项的存在会覆盖该选项的使用。在路径 {pid} 中将被当前进程的 PID 替换。
client_socket_timeout = 900	整数值	客户端连接的套接字操作的超时。如果传入的连接闲置了这个秒数，它将被关闭。值 0 表示等待永久。
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
control_exchange = openstack	字符串值	默认交换，哪些主题在哪些方面有范围。可以被 <code>transport_url</code> 选项指定的交换名称覆盖。
db_auto_create = False	布尔值	在服务启动时创建 Barbican 数据库。

配置选项 = 默认值	类型	Description
debug = False	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。
default_limit_paging = 10	整数值	<i>限制</i> 分页 URL 参数的默认页面大小。
default_log_levels = ['amqp=WARN', 'amqplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 log_config_append，则忽略此选项。
executor_thread_pool_size = 64	整数值	当 executor 为 threading 或 eventlet 时，executor 线程池的大小。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
host_href = http://localhost:9311	字符串值	主机名，用于 HATEOAS 式参考。请注意，这是客户端用来与此服务通信的负载均衡端点。如果部署希望从 wsgi 请求中派生主机，则将此设置为空白。需要为空白来覆盖默认配置值，即 http://localhost:9311
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。

配置选项 = 默认值	类型	Description
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 <code>log-date-format</code> ）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime)s.%(msecs)03d.%(process)d.%(levelname)s.%(name)s.%(request_id)s.%(user_identity)s.%(instance)s.%(message)s	字符串值	使用上下文格式化用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_debug_format_sufffix = %(funcName)s.%(pathname)s.%(lineno)d	字符串值	当消息的日志记录级别为 <code>DEBUG</code> 时，要附加到日志消息的其他数据。 <code>oslo_log.formatters.ContextFormatter</code> 使用

配置选项 = 默认值	类型	Description
logging_default_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_format = %(user) s %(tenant) s %(domain) s % (user_domain) s % (project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_allowed_request_size_in_bytes = 25000	整数值	针对 barbican-api 的最大允许 http 请求大小。
max_allowed_secret_in_bytes = 20000	整数值	允许的最大 secret 大小（以字节为单位）。
max_header_line = 16384	整数值	要接受的最大消息标头大小。在使用大令牌时可能需要增加 max_header_line（通常是在 keystone 配置为使用具有大型服务目录的 PKI 令牌时生成的）。
max_limit_paging = 100	整数值	<i>限制</i> 分页 URL 参数的最大页面大小。
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size", 则忽略这个选项。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。

配置选项 = 默认值	类型	Description
rpc_conn_pool_size = 30	整数值	RPC 连接池的大小。
rpc_ping_enabled = False	布尔值	添加端点以回答 ping 调用。端点名为 <code>oslo_rpc_server_ping</code>
rpc_response_timeout = 60	整数值	从调用等待响应的秒数。
run_external_periodic_tasks = True	布尔值	某些定期任务可以在单独的进程中运行。我们是否应该在此处运行？
sql_connection = sqlite:///barbican.sqlite	字符串值	用于参考实现 registry 服务器的 sqlalchemy 连接字符串。任何有效的 SQLAlchemy 连接字符串都是正常的。请参阅： http://www.sqlalchemy.org/docs/05/reference/sqlalchemy/connections.html#sqlalchemy.create_engine 。注意：对于绝对地址，请在 <code>sqlite</code> 后使用 <code>////</code> 斜杠：
sql_idle_timeout = 3600	整数值	SQLAlchemy 应该重新建立与数据库的连接的时间（以秒为单位）。MySQL 使用默认的 <code>wait_timeout</code> 8 小时，它将丢弃闲置连接。这可能导致 MySQL <code>Gone Away</code> 例外。如果注意到这个值，您可以降低这个值来确保在 MySQL 可以丢弃连接前 SQLAlchemy 重新连接。
sql_max_retries = 60	整数值	启动期间数据库连接重试的最大数量。设置为 -1 以指定无限重试计数。
sql_pool_class = QueuePool	字符串值	接受从 <code>sqlalchemy.pool</code> 模块导入的类，并处理为您构建池的详细信息。如果注释掉，SQLAlchemy 将根据数据库划分来选择。其他选项包括 <code>QueuePool</code> （用于 SQLAlchemy 连接）和 <code>NullPool</code> （禁用的 SQLAlchemy 连接管理）。详情请查看 http://docs.sqlalchemy.org/en/latest/core/pooling.html
sql_pool_logging = False	布尔值	在日志中显示 SQLAlchemy 池相关的调试输出（如果指定，则显示 DEBUG 日志级别输出）。
sql_pool_max_overflow = 10	整数值	SQLAlchemy 使用的池的最大溢出大小。当签出的连接数量达到 <code>sql_pool_size</code> 中设置的大小时，会将额外的连接返回到这个限制。然后，池允许的同时连接总数为 <code>sql_pool_size + sql_pool_max_overflow</code> 。可以设置为 -1 以表示没有溢出限制，因此不会将限制放在并发连接总数中。注释掉允许 SQLAlchemy 选择默认值。

配置选项 = 默认值	类型	Description
sql_pool_size = 5	整数值	SQLAlchemy 使用的池大小。这是永久保留在池中的最大连接数。可以设置为 0 以表示没有大小限制。要禁用池，请使用带有 <code>sql_pool_class</code> 的 <code>NullPool</code> 。注释掉允许 SQLAlchemy 选择默认值。
sql_retry_interval = 1	整数值	重新尝试打开 SQL 连接之间的间隔。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 <code>log_config_append</code> ，则忽略此选项。
tcp_keepidle = 600	整数值	为每个服务器套接字设置 <code>TCP_KEEPIDLE</code> 的值（以秒为单位）。在 OS X 上不支持。
transport_url = rabbit://	字符串值	<p>用于连接到消息传递后端的网络地址和可选用户凭证，采用 URL 格式。预期格式为：</p> <pre>driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query</pre> <p>示例：<code>rabbit://rabbitmq:password@127.0.0.1:5672//</code></p> <p>有关 URL 中字段的详情，请查看 <code>oslo_messaging.TransportURL</code> 文档 https://docs.openstack.org/oslo.messaging/latest/reference/transport.html</p>
use-journal = False	布尔值	为日志记录启用 <code>journald</code> 。如果在 <code>systemd</code> 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 <code>log_config_append</code> ，此选项将被忽略。
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
use-syslog = False	布尔值	使用 <code>syslog</code> 进行日志记录。现有的 <code>syslog</code> 格式是 <code>DEPRECATED</code> ，稍后将更改为遵循 <code>RFC5424</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 <code>log_config_append</code> ，则忽略此选项。
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 <code>log_file</code> 选项并使用 Linux 平台时，才有意义。如果设置了 <code>log_config_append</code> ，则忽略此选项。

配置选项 = 默认值	类型	Description
wsgi_default_pool_size = 100	整数值	wsgi 使用的 greenthreads 池的大小
wsgi_keep_alive = True	布尔值	如果为 False，请明确关闭客户端套接字连接。
wsgi_log_format = %(client_ip) s "%(request_line) s" status: %(status_code) s len: %(body_length) s time: %(wall_seconds).7f	字符串值	用作模板用于生成日志行的 python 格式字符串。以下值可以格式化到其中：client_ip, date_time, request_line, status_code, body_length, wall_seconds。
wsgi_server_debug = False	布尔值	如果服务器应该将异常回溯发送到客户端 500 错误，则为 true。如果为 False，服务器将响应空正文。

1.1.2. certificate

下表概述了 `/etc/barbican/barbican.conf` 文件中的 **[certificate]** 组下可用的选项。

表 1.1. certificate

配置选项 = 默认值	类型	Description
enabled_certificate_plugins = ['simple_certificate']	多值	要加载的证书插件列表。
namespace = barbican.certificate.plugin	字符串值	扩展命名空间以搜索插件。

1.1.3. certificate_event

下表概述了 `/etc/barbican/barbican.conf` 文件中的 **[certificate_event]** 组下可用的选项。

表 1.2. certificate_event

配置选项 = 默认值	类型	Description
enabled_certificate_event_plugins = ['simple_certificate_event']	多值	要加载的证书插件列表。
namespace = barbican.certificate.event.plugin	字符串值	扩展命名空间以搜索事件插件。

1.1.4. CORS

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[cors]` 组下可用的选项。

表 1.3. CORS

配置选项 = 默认值	类型	Description
<code>allow_credentials = True</code>	布尔值	指明实际请求可以包含用户凭证
<code>allow_headers = ['X-Auth-Token', 'X-Openstack-Request-Id', 'X-Project-Id', 'X-Identity-Status', 'X-User-Id', 'X-Storage-Token', 'X-Domain-Id', 'X-User-Domain-Id', 'X-Project-Domain-Id', 'X-Roles']</code>	列表值	指定在实际请求过程中可以使用哪些标头字段名称。
<code>allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH']</code>	列表值	指明在实际请求期间可以使用哪些方法。
<code>allowed_origin = None</code>	列表值	指明此资源是否可以与请求"origin"标头中收到的域共享。格式： <code>"<protocol>://<host>[:<port>]"</code> ，没有尾部斜杠。示例： https://horizon.example.com
<code>expose_headers = ['X-Auth-Token', 'X-Openstack-Request-Id', 'X-Project-Id', 'X-Identity-Status', 'X-User-Id', 'X-Storage-Token', 'X-Domain-Id', 'X-User-Domain-Id', 'X-Project-Domain-Id', 'X-Roles']</code>	列表值	指明哪些标头可以安全地公开给 API。默认为 HTTP Simple Headers。
<code>max_age = 3600</code>	整数值	CORS preflight 请求的最大缓存期限。

1.1.5. crypto

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[crypto]` 组下可用的选项。

表 1.4. crypto

配置选项 = 默认值	类型	Description
<code>enabled_crypto_plugins = ['simple_crypto']</code>	多值	要载入的加密插件列表。

配置选项 = 默认值	类型	Description
namespace = barbican.crypto.plugin	字符串值	扩展命名空间以搜索插件。

1.1.6. dogtag_plugin

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[dogtag_plugin]` 组下可用的选项。

表 1.5. dogtag_plugin

配置选项 = 默认值	类型	Description
auto_approved_profiles = caServerCert	字符串值	自动批准的注册配置文件列表
ca_expiration_time = 1	整数值	CA 条目过期的时间
dogtag_host = localhost	字符串值	Dogtag 实例的主机名
dogtag_port = 8443	端口值	Dogtag 实例的端口
nss_db_path = /etc/barbican/alias	字符串值	NSS 证书数据库的路径
nss_password = None	字符串值	NSS 证书数据库的密码
pem_path = /etc/barbican/kra_admin_cert.pem	字符串值	PEM 文件的路径进行验证
plugin_name = Dogtag KRA	字符串值	用户友好插件名称
plugin_working_dir = /etc/barbican/dogtag	字符串值	Dogtag 插件的工作目录
retries = 3	整数值	在存储或生成 secret 时重试
simple_cmc_profile = caOtherCert	字符串值	简单 CMC 请求的配置集

1.1.7. keystone_authtoken

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[keystone_authtoken]` 组下可用的选项。

表 1.6. keystone_authtoken

配置选项 = 默认值	类型	Description
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
auth_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点以进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。 这个选项已弃用，现在使用 <code>www_authenticate_uri</code>，并将在 S 发行版本中删除。弃用自：Queens *reason:* <code>auth_uri</code> 选项已弃用，现在使用 <code>www_authenticate_uri</code>，并将在 S 发行版本中删除。
auth_version = None	字符串值	Identity API 端点的 API 版本。
cache = None	字符串值	存储 Swift 缓存对象的请求环境密钥。当使用 Swift 缓存部署 <code>auth_token</code> 中间件时，请使用此选项使中间件共享具有 <code>swift</code> 的缓存后端。否则，使用 <code>memcached_servers</code> 选项。
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。默认为系统 CA。
certFile = None	字符串值	如果身份服务器需要客户端证书，则需要此项
delay_auth_decision = False	布尔值	不要在中间件内处理授权请求，而是将授权决策委派给下游 WSGI 组件。
enforce_token_bind = permissive	字符串值	用于控制令牌绑定的使用和类型。可以设置为 "disabled" 以不检查令牌绑定。如果绑定类型是已知服务器的绑定类型，则验证绑定信息，并忽略它（如 "permissive"），但如果绑定类型未知，则令牌将被拒绝。"必需"令牌绑定需要允许任何形式的令牌绑定。最后，令牌中必须存在的绑定方法名称。
http_connect_timeout = None	整数值	请求超时值与身份 API 服务器通信。
http_request_max_retries = 3	整数值	与身份 API 服务器通信时，我们尝试重新连接的次数。
include_service_catalog = True	布尔值	（可选）指示是否设置 X-Service-Catalog 标头。如果为 False，则中间件不会要求在令牌验证时请求服务目录，且不会设置 X-Service-Catalog 标头。

配置选项 = 默认值	类型	Description
insecure = False	布尔值	验证 HTTPS 连接。
interface = internal	字符串值	用于 Identity API 端点的接口。有效值为 "public", "internal" (默认) 或 "admin"。
keyfile = None	字符串值	如果身份服务器需要客户端证书, 则需要此项
memcache_pool_conn_get_timeout = 10	整数值	(可选) 操作等待从池中获取 memcached 客户端连接的秒数。
memcache_pool_dead_retry = 300	整数值	(可选) 在再次尝试前 memcached 服务器被视为 dead 的秒数。
memcache_pool_maxsize = 10	整数值	(可选) 到每个 memcached 服务器的最大打开连接数。
memcache_pool_socket_timeout = 3	整数值	(可选) 与 memcached 服务器通信的套接字超时 (以秒为单位)。
memcache_pool_unused_timeout = 60	整数值	(可选) 与 memcached 的连接在池关闭前未使用的秒数。
memcache_secret_key = None	字符串值	(可选, 如果定义了 memcache_security_strategy) 此字符串用于密钥派生。
memcache_security_strategy = None	字符串值	(可选) 如果定义, 则指示令牌数据是否应进行身份验证或加密。如果 MAC, 令牌数据在缓存中进行身份验证 (使用 HMAC)。如果 ENCRYPT, 令牌数据会在缓存中加密并进行身份验证。如果该值不是这些选项之一或为空, 则 auth_token 会在初始化时引发异常。
memcache_use_advanced_pool = False	布尔值	(可选) 使用 advanced (事件安全) memcached 客户端池。高级池只能在 python 2.x 下工作。
memcached_servers = None	列表值	(可选) 指定要用于缓存的 memcached 服务器列表。如果未定义, 令牌将会被缓存在进程中。
region_name = None	字符串值	找到身份服务器的区域。
service_token_roles = ['service']	列表值	需要出现在服务令牌中的角色。服务令牌可以请求可以使用过期的令牌, 因此此检查应严格控制只有实际服务应发送此令牌。此处的角色作为 ANY 检查应用, 因此此列表中的任何角色都必须存在。为了向后兼容的原因, 这目前只会影响 allow_expired 检查。

配置选项 = 默认值	类型	Description
service_token_roles_required = False	布尔值	为了向后兼容的原因，我们必须允许有效的服务令牌传递，但不允许将 service_token_roles 检查视为有效。在以后的发行版本中设置此 true 将成为默认设置，并应启用（如果可能）。
service_type = None	字符串值	服务的名称或类型（在服务目录中出现）。这用于验证具有受限访问规则的令牌。
token_cache_time = 300	整数值	为了防止过度努力验证令牌，中间件会缓存之前可预见的令牌（以秒为单位）。设置为 -1 可完全禁用缓存。
www_authenticate_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。

1.1.8. keystone_notifications

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[keystone_notifications]` 组下可用的选项。

表 1.7. keystone_notifications

配置选项 = 默认值	类型	Description
allow_requeue = False	布尔值	true 在通知处理错误时启用 requeue 功能。仅在底层传输支持此功能时启用此功能。
control_exchange = keystone	字符串值	默认交换，哪些主题在哪些方面有范围。可以被 transport_url 选项指定的交换名称覆盖。
enable = False	布尔值	true 启用 keystone 通知监听程序功能。
pool_name = None	字符串值	通知监听程序的池名称。把它设置为不同的值将允许 barbican 通知监听程序从主题接收其自身所有消息的副本，而不会干扰侦听同一主题的其他服务。此功能仅受某些 oslo.messaging 后端（由 rabbitmq 的 particular 中）支持，对于那些最好使用这个功能，而不是单独的 barbican 通知主题。
thread_pool_size = 10	整数值	定义用于通知服务器处理功能的最大线程数量。

配置选项 = 默认值	类型	Description
topic = notification	字符串值	Keystone 通知队列主题名称。此名称需要匹配 Keystone 部署的 <i>notification_topics</i> 配置中提到的其中一个值，例如 <i>notification_topics=notifications, barbican_notifications</i> 。Multiple 服务器可以侦听一个主题，消息将以轮循方式分配给其中一个服务器。这就是 Barbican 服务应拥有自己的专用通知队列的原因，以便它接收所有 Keystone 通知。或者，如果所选的 <i>oslo.messaging</i> 后端支持侦听器池（如 <i>rabbitmq</i> ），则应首选设置非默认 <i>pool_name</i> 选项。
version = 1.0	字符串值	通过通知调用的任务版本

1.1.9. kmip_plugin

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[kmip_plugin]` 组下可用的选项。

表 1.8. kmip_plugin

配置选项 = 默认值	类型	Description
ca_certs = None	字符串值	串联“认证机构”证书的文件路径
certFile = None	字符串值	到本地客户端证书的文件路径
host = localhost	字符串值	KMIP 服务器的地址
keyfile = None	字符串值	到本地客户端证书密钥文件的文件路径
password = None	字符串值	使用 KMIP 服务器进行身份验证的密码
pkcs1_only = False	布尔值	只支持非对称密钥的 PKCS#1 编码
plugin_name = KMIP HSM	字符串值	用户友好插件名称
port = 5696	端口值	KMIP 服务器的端口
ssl_version = PROTOCOL_TLSv1_2	字符串值	SSL 版本，映射到模块的 <i>ssl</i> 常数
username = None	字符串值	使用 KMIP 服务器进行身份验证的用户名

1.1.10. oslo_messaging_amqp

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[oslo_messaging_amqp]` 组下可用的选项。

表 1.9. oslo_messaging_amqp

配置选项 = 默认值	类型	Description
addressing_mode = dynamic	字符串值	表示驱动程序使用的寻址模式。允许的值： <i>legacy</i> - 使用旧的不可路由寻址； <i>routable</i> - 使用可路由地址； <i>dynamic</i> - 如果消息总线不支持路由，使用 <i>legacy</i> ，如果支持，使用 <i>routable</i> 。
anycast_address = anycast	字符串值	在发送到一组消费者时附加到地址前缀。消息总线用来识别应在消费者之间以轮循方式发送的消息。
broadcast_prefix = broadcast	字符串值	广播到所有服务器时使用的地址前缀
connection_retry_backoff = 2	整数值	每次失败故障转移尝试后，将 <code>connection_retry_interval</code> 增加到这秒数。
connection_retry_interval = 1	整数值	尝试重新连接前暂停的秒数。
connection_retry_interval_max = 30	整数值	<code>connection_retry_interval</code> + <code>connection_retry_backoff</code> 的最大限制
container_name = None	字符串值	AMQP 容器的名称必须全局唯一。默认为生成的 UUID
default_notification_exchange = None	字符串值	通知地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_notification_exchange if set else control_exchange if set else <i>notify</i>
default_notify_timeout = 30	整数值	发送通知消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_reply_retry = 0	整数值	重新发送回复消息的最大尝试次数，因为可恢复错误而失败。
default_reply_timeout = 30	整数值	rpc 回复消息发送的截止时间。
default_rpc_exchange = None	字符串值	RPC 地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_rpc_exchange if set other control_exchange if set else <i>rpc</i>
default_send_timeout = 30	整数值	rpc cast 或调用消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_sender_link_timeout = 600	整数值	调度闲置发送者链接清除的持续时间。过期后分离链接。

配置选项 = 默认值	类型	Description
group_request_prefix = unicast	字符串值	发送到组中任何服务器时的地址前缀
idle_timeout = 0	整数值	不活跃连接的超时时间（以秒为单位）
link_retry_delay = 10	整数值	重新连接 AMQP 1.0 链接之间的暂停时间，该链接因为可恢复的错误而失败。
multicast_address = multicast	字符串值	在发送异常消息时，附加到地址前缀。消息总线用来识别 fanout 消息。
notify_address_prefix = openstack.org/om/notify	字符串值	所有生成的通知地址的地址前缀
notify_server_credit = 100	整数值	传入通知消息的窗口大小
pre_settled = ['rpc-cast', 'rpc-reply']	多值	发送此类型的消息预先组装。预先设置的消息不会接收来自 peer 的确认信息。请注意：如果发送失败，则预先设置的消息可能会被静默丢弃。允许的值： <i>rpc-call</i> - 发送 RPC Calls <i>pre-settled</i> <i>rpc-reply</i> - 发送 RPC Replies <i>pre-settled</i> <i>rpc-cast</i> - Send RPC Casts <i>pre-settled</i> <i>notify</i> - Send Notifications <i>pre-settled</i>
pseudo_vhost = True	布尔值	为不支持虚拟主机的消息总线启用虚拟主机支持（如 qpidd）。当设置为 true 时，虚拟主机名将添加到所有消息总线地址中，从而有效地为每个虚拟主机 <i>创建专用子网</i> 。如果消息总线支持使用 AMQP 1.0 Open performative 中的 <i>hostname</i> 字段作为虚拟主机的名称，设置为 False。
reply_link_credit = 200	整数值	传入 RPC 消息的窗口大小。
rpc_address_prefix = openstack.org/om/rpc	字符串值	所有生成的 RPC 地址的地址前缀
rpc_server_credit = 100	整数值	传入 RPC Request 消息的窗口大小
<code>`sas_config_dir = `</code>	字符串值	包含 SASL 配置的目录路径
<code>`sas_config_name = `</code>	字符串值	配置文件名称（不带 .conf 后缀）
<code>`sas_default_realm = `</code>	字符串值	如果用户名中没有 realm，则要使用的 SASL 域
<code>`sas_mechanisms = `</code>	字符串值	可接受的 SASL 机制的以空格分隔的列表

配置选项 = 默认值	类型	Description
server_request_prefix = exclusive	字符串值	发送到特定服务器时使用的地址前缀
ssl = False	布尔值	尝试通过 SSL 连接。如果没有给出其他与 ssl 相关的参数，它将使用系统的 CA-bundle 来验证服务器证书。
<code>`ssl_ca_file = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_cert_file = `</code>	字符串值	自我标识用于客户端身份验证的证书 PEM 文件
<code>`ssl_key_file = `</code>	字符串值	用于为 ssl_cert_file 证书签名的私钥 PEM 文件（可选）
ssl_key_password = None	字符串值	用于解密 ssl_key_file 的密码（如果加密）
ssl_verify_vhost = False	布尔值	默认情况下，SSL 检查服务器证书中的名称与 transport_url 中的主机名匹配。在某些配置中，最好使用虚拟主机名。例如，如果服务器使用 Server Name Indication TLS extension (rfc6066) 来为每个虚拟主机提供证书。如果服务器的 SSL 证书使用虚拟主机名称而不是 DNS 名称，则将 ssl_verify_vhost 设置为 True。
trace = False	布尔值	debug : 将 AMQP 帧转储到 stdout
unicast_address = unicast	字符串值	在发送到特定 RPC/Notification 服务器时，附加到地址前缀。消息总线用来识别发送到单个目的地的消息。

1.1.11. oslo_messaging_kafka

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[oslo_messaging_kafka]` 组下可用的选项。

表 1.10. oslo_messaging_kafka

配置选项 = 默认值	类型	Description
compression_codec = none	字符串值	producer 生成的所有数据的压缩 codec。如果没有设置，则不会使用压缩。请注意，这允许的值取决于 kafka 版本
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟

配置选项 = 默认值	类型	Description
consumer_group = oslo_messaging_consumer	字符串值	Kafka 消费者的组 ID。一个组中的消费者将协调消息消耗
enable_auto_commit = False	布尔值	启用异步消费者提交
kafka_consumer_timeout = 1.0	浮点值	Kafka 用户的默认超时
kafka_max_fetch_bytes = 1048576	整数值	Kafka consumer 的最大获取字节数
max_poll_records = 500	整数值	轮询调用中返回的最大记录数
pool_size = 10	整数值	Kafka Consumers 的池大小
producer_batch_size = 16384	整数值	producer async 发送的批处理大小
producer_batch_timeout = 0.0	浮点值	以秒为单位的 KafkaProducer 批处理延迟上限
sasl_mechanism = PLAIN	字符串值	当安全协议是 SASL 时的机制
security_protocol = PLAINTEXT	字符串值	用于与代理通信的协议
<code>`ssl_cafile = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_client_cert_file = `</code>	字符串值	用于身份验证的客户端证书 PEM 文件。
<code>`ssl_client_key_file = `</code>	字符串值	用于身份验证的客户端密钥 PEM 文件。
<code>`ssl_client_key_password = `</code>	字符串值	用于身份验证的客户端密钥密码文件。

1.1.12. oslo_messaging_notifications

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[oslo_messaging_notifications]` 组下可用的选项。

表 1.11. oslo_messaging_notifications

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
driver = []	多值	处理发送通知的驱动程序。可能的值有 messaging, messagingv2, routing, log, test, noop
retry = -1	整数值	重新发送因为可恢复错误而发送失败的通知消息的最大尝试次数。0 - 不重试, -1 - indefinite
topics = ['notifications']	列表值	用于 OpenStack 通知的 AMQP 主题。
transport_url = None	字符串值	代表用于通知的消息传递驱动程序的 URL。如果没有设置, 我们回退到用于 RPC 的同一配置。

1.1.13. oslo_messaging_rabbit

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[oslo_messaging_rabbit]` 组下可用的选项。

表 1.12. oslo_messaging_rabbit

配置选项 = 默认值	类型	Description
amqp_auto_delete = False	布尔值	AMQP 中的自动删除队列。
amqp_durable_queues = False	布尔值	在 AMQP 中使用持久队列。
direct_mandatory_flag = True	布尔值	(DEPRECATED)为直接发送启用/禁用 RabbitMQ 强制标记。直接发送用作回复, 因此在客户端队列不存在时引发 MessageUndeliverable 异常。MessageUndeliverable 异常将用于循环使发送者恢复。此标志已弃用, 且无法停用此功能
enable_cancel_on_failover = False	布尔值	启用 x-cancel-on-ha-failover 标志, 使 rabbitmq 服务器将取消并通知 consumerswhen 队列停机
heartbeat_in_pthread = False	布尔值	默认情况下, 通过原生 python 线程运行健康检查心跳线程。如果此选项等于 False, 则健康检查 heartbeat 将继承父进程的执行模型。例如, 如果父进程使用 eventlet/greenlet 对 stdlib 进行补丁, 则心跳将通过绿色线程运行。这个选项应该只针对 wsgi 服务设置为 True。
heartbeat_rate = 2	整数值	heartbeat_timeout_threshold 期间的我们检查心跳的频率。
heartbeat_timeout_threshold = 60	整数值	如果 heartbeat 的 keep-alive 失败(0 禁用 heartbeat), 则 Rabbit 代理被视为 down 的秒数。

配置选项 = 默认值	类型	Description
kombu_compression = None	字符串值	EXPERIMENTAL : Possible 值为 : gzip、bz2。如果没有设置压缩, 则不会使用。这个选项可能在以后的版本中不可用。
kombu_failover_strategy = round-robin	字符串值	确定当当前连接到的 RabbitMQ 节点不可用时如何选择下一个 RabbitMQ 节点。仅在配置中提供多个 RabbitMQ 节点时生效。
kombu_missing_consumer_retry_timeout = 60	整数值	在放弃发送其回复前等待缺少的客户端的时间。这个值不应比 <code>rpc_response_timeout</code> 长。
kombu_reconnect_delay = 1.0	浮点值	响应 AMQP 消费者取消通知前等待多久。
rabbit_ha_queues = False	布尔值	尝试在 RabbitMQ 中使用 HA 队列(<code>x-ha-policy: all</code>)。如果更改这个选项, 您必须擦除 RabbitMQ 数据库。在 RabbitMQ 3.0 中, 在声明队列时, <code>x-ha-policy</code> 参数不再控制队列镜像。如果您只想确保所有队列(带有自动生成的名称的队列除外)都在所有节点上进行镜像, 请运行: <code>"rabbitmqctl set_policy HA "(?!amq\.)gem {"ha-mode": "all"}"</code>
rabbit_interval_max = 30	整数值	RabbitMQ 连接重试的最大间隔。默认值为 30 秒。
rabbit_login_method = AMQPLAIN	字符串值	RabbitMQ 登录方法。
rabbit_qos_prefetch_count = 0	整数值	指定要预抓取的消息数量。设置为零可允许无限消息。
rabbit_retry_backoff = 2	整数值	连接到 RabbitMQ 时重试之间的 backoff 时长。
rabbit_retry_interval = 1	整数值	使用 RabbitMQ 重试连接的频率。
rabbit_transient_queues_ttl = 1800	整数值	正整数代表队列 TTL (<code>x-expires</code>)的持续时间(以秒为单位)。TTL 期间未使用的队列将被自动删除。参数仅影响 <code>reply</code> 和 <code>fanout</code> 队列。
ssl = False	布尔值	通过 SSL 连接。
<code>`ssl_ca_file = `</code>	字符串值	SSL 证书颁发机构文件(仅在启用了 SSL 时评估)。
<code>`ssl_cert_file = `</code>	字符串值	SSL 证书文件(仅在启用了 SSL 时评估)。
<code>`ssl_key_file = `</code>	字符串值	SSL 密钥文件(仅在启用了 SSL 时 valid)。

配置选项 = 默认值	类型	Description
<code>`ssl_version = `</code>	字符串值	要使用的 SSL 版本（仅在启用了 SSL 时评估）。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1_1 和 TLSv1_2 可能在某些发行版中提供。

1.1.14. oslo_middleware

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[oslo_middleware]` 组下可用的选项。

表 1.13. oslo_middleware

配置选项 = 默认值	类型	Description
<code>enable_proxy_headers_parsing = False</code>	布尔值	应用程序是否在代理后面。这决定了中间件是否应该解析标头。

1.1.15. oslo_policy

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[oslo_policy]` 组下可用的选项。

表 1.14. oslo_policy

配置选项 = 默认值	类型	描述
<code>enforce_new_defaults = False</code>	布尔值	这个选项控制在评估策略时是否使用旧的弃用默认值。如果为 True ，则不会评估旧的已弃用的默认值。这意味着，如果旧默认值允许任何现有令牌，但不允许使用新默认值，它将被禁止。我们鼓励启用此标志以及 <code>enforce_scope</code> 标志，以便您可以同时获得新默认值和 <code>scope_type</code> 的好处。
<code>enforce_scope = False</code>	布尔值	此选项控制在评估策略时是否强制执行范围。如果为 True ，则请求中使用的令牌范围将与被强制执行策略的 <code>scope_types</code> 进行比较。如果范围不匹配，则引发 <code>InvalidScope</code> 异常。如果为 False ，则会记录一条信息，告知操作员使用了不匹配的范围调用策略。
<code>policy_default_rule = default</code>	字符串值	默认规则。未找到请求的规则时强制执行。
<code>policy_dirs = ['policy.d']</code>	多值	存储策略配置文件的目录。它们可以相对于 <code>config_dir</code> 选项或绝对路径定义的搜索路径中的任何目录。由 <code>policy_file</code> 定义的文件必须存在，才能搜索这些目录。缺少或空目录将被忽略。
<code>policy_file = policy.yaml</code>	字符串值	将角色映射到给定服务权限的文件的相对或绝对路径。必须使用与配置文件设置相关的相对路径。

配置选项 = 默认值	类型	描述
remote_content_type = application/x-www-form-urlencoded	字符串值	用于发送和接收基于 REST 策略检查的数据的内容类型
remote_ssl_ca_cert_file = None	字符串值	基于 REST 的策略检查的 ca cert 文件的绝对路径
remote_ssl_client_cert_file = None	字符串值	基于 REST 的策略检查的客户端证书的绝对路径
remote_ssl_client_key_file = None	字符串值	绝对路径客户端密钥文件 REST 基于策略检查
remote_ssl_verify_server_cert = False	布尔值	基于 REST 的策略检查的服务器身份验证

1.1.16. p11_crypto_plugin

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[p11_crypto_plugin]` 组下可用的选项。

表 1.15. p11_crypto_plugin

配置选项 = 默认值	类型	Description
aes_gcm_generate_iv = True	布尔值	为 CKM_AES_GCM 机制生成 IVs。
always_set_cka_sensitive = True	布尔值	始终设置 CKA_SENSITIVE=CK_TRUE 包括 CKA_EXTRACTABLE=CK_TRUE 密钥。
encryption_mechanism = CKM_AES_CBC	字符串值	Secret 加密机制
hmac_key_type = CKK_AES	字符串值	HMAC 密钥类型
hmac_keygen_mechanism = CKM_AES_KEY_GEN	字符串值	HMAC 密钥生成算法
hmac_keywrap_mechanism = CKM_SHA256_HMAC	字符串值	HMAC 键 wrap 机制
hmac_label = None	字符串值	Master HMAC Key 标签 (如 HSM 中存储的)
library_path = None	字符串值	供应商 PKCS11 库的路径

配置选项 = 默认值	类型	Description
login = None	字符串值	登录到 PKCS11 会话的密码
mkek_label = None	字符串值	Master KEK 标签（存储在 HSM 中）
mkek_length = None	整数值	master KEK 长度（以字节为单位）。
os_locking_ok = False	布尔值	初始化 PKCS facilities 客户端库时，启用 CKF_OS_LOCKING_OK 标志。
pkek_cache_limit = 100	整数值	项目 KEK Cache Item Limit
pkek_cache_ttl = 900	整数值	项目 KEK Cache Time To Live，以秒为单位
pkek_length = 32	整数值	项目 KEK 长度（以字节为单位）。
plugin_name = PKCS11 HSM	字符串值	用户友好插件名称
rw_session = True	布尔值	读/写会话的标记
<code>`seed_file = `</code>	字符串值	用于拉取有关 seeding RNG 的熵的文件
seed_length = 32	整数值	从文件读取的数据量用于 seed
slot_id = 1	整数值	（可选）包含要使用的令牌设备的 HSM Slot ID。
token_label = None	字符串值	DEPRECATED：改为使用 token_labels。用于识别要使用的令牌的令牌标签。
token_labels = None	列表值	要使用的一个或多个令牌的标签列表。通常，这是一个标签，但有些 HSM 设备可能需要多个标签用于负载均衡或高可用性配置。
token_serial_number = None	字符串值	用于识别要使用的令牌的令牌序列号。

1.1.17. queue

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[queue]` 组下可用的选项。

表 1.16. queue

配置选项 = 默认值	类型	Description
asynchronous_workers = 1	整数值	异步 worker 进程数量

配置选项 = 默认值	类型	Description
enable = False	布尔值	true 启用排队, False 同步调用 worker
namespace = barbican	字符串值	队列命名空间
server_name = barbican.queue	字符串值	RPC 任务处理服务器的服务器名称
topic = barbican.workers	字符串值	队列主题名称
version = 1.1	字符串值	通过队列调用的任务版本

1.1.18. quotas

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[quotas]` 组下可用的选项。

表 1.17. quotas

配置选项 = 默认值	类型	Description
quota_cas = -1	整数值	每个项目允许的 CA 数量
quota_consumers = -1	整数值	每个项目允许的消费者数量
quota_containers = -1	整数值	每个项目允许的容器数量
quota_orders = -1	整数值	每个项目允许的订购数
quota_secrets = -1	整数值	每个项目允许的 secret 数量

1.1.19. retry_scheduler

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[retry_scheduler]` 组下可用的选项。

表 1.18. retry_scheduler

配置选项 = 默认值	类型	Description
initial_delay_seconds = 10.0	浮点值	启动重试调度程序前要等待的秒数 (浮点数)
periodic_interval_max_seconds = 10.0	浮点值	在定期调度事件之间等待的秒数 (浮点)

1.1.20. secretstore

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[secretstore]` 组下可用的选项。

表 1.19. secretstore

配置选项 = 默认值	类型	Description
enable_multiple_secret_stores = False	布尔值	启用多个 secret 存储插件后端的标志。默认为 False
enabled_secretstore_plugins = ['store_crypto']	多值	要加载的机密存储插件列表。
namespace = barbican.secretstore.plugin	字符串值	扩展命名空间以搜索插件。
stores_lookup_suffix = None	列表值	用于查找支持多个后端支持的插件的后缀列表。

1.1.21. simple_crypto_plugin

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[simple_crypto_plugin]` 组下可用的选项。

表 1.20. simple_crypto_plugin

配置选项 = 默认值	类型	Description
kek = dGhpcnR5X3R3b19ieXRI X2tleWJsYWhibGFoYmxh aGg=	字符串值	Simple Crypto Plugin 使用的密钥加密密钥
plugin_name = Software Only Crypto	字符串值	用户友好插件名称

1.1.22. snakeoil_ca_plugin

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[snakeoil_ca_plugin]` 组下可用的选项。

表 1.21. snakeoil_ca_plugin

配置选项 = 默认值	类型	Description
ca_cert_chain_path = None	字符串值	CA 证书链文件的路径
ca_cert_key_path = None	字符串值	CA 证书密钥文件的路径
ca_cert_path = None	字符串值	CA 证书文件的路径

配置选项 = 默认值	类型	Description
ca_cert_pkcs7_path = None	字符串值	CA 链 pkcs7 文件的路径
subca_cert_key_directory = /etc/barbican/snakeoil-cas	字符串值	为 subcas 保存证书/密钥的目录

1.1.23. ssl

下表概述了 `/etc/barbican/barbican.conf` 文件中的 `[ssl]` 组下可用的选项。

表 1.22. ssl

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	用于验证连接客户端的 CA 证书文件。
cert_file = None	字符串值	安全启动服务器时要使用的证书文件。
密码 = None	字符串值	设置可用密码列表。值应该是 OpenSSL 密码列表格式的字符串。
key_file = None	字符串值	安全启动服务器时使用的私钥文件。
version = None	字符串值	要使用的 SSL 版本（仅在启用了 SSL 时评估）。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1_1 和 TLSv1_2 可能在某些发行版中提供。

第 2 章 CINDER

下面的章节包含有关 `cinder` 服务中的配置选项的信息。

2.1. CINDER.CONF

本节包含 `/etc/cinder/cinder.conf` 文件的选项。

2.1.1. DEFAULT

下表概述了 `/etc/cinder/cinder.conf` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	描述
<code>acs5000_copy_interval = 5</code>	整数值	当卷复制任务进入时，刷新卷状态间隔
<code>acs5000_volpool_name = ['pool01']</code>	列表值	卷存储系统存储池的逗号分隔列表。
<code>allocated_capacity_weight_multiplier = -1.0</code>	浮点值	用于权衡分配的容量的倍数。正数意味着堆栈与分散。
<code>allow_availability_zone_fallback = False</code>	布尔值	如果请求的 Cinder 可用性区域不可用，请回退到 <code>default_availability_zone</code> 的值，然后是 <code>storage_availability_zone</code> ，而不是 <code>fail</code> 。
<code>allow_compression_on_image_upload = False</code>	布尔值	上传时用于镜像压缩的策略。默认为禁止压缩。
<code>allowed_direct_url_schemes = []</code>	列表值	可以直接通过 <code>direct_url</code> 下载的 url 方案列表。目前支持的方案： <code>[file, cinder]</code> 。
<code>api_paste_config = api-paste.ini</code>	字符串值	api 服务的 <code>paste.deploy</code> 配置的文件名
<code>api_rate_limit = True</code>	布尔值	启用或禁用 API 的速率限制。
<code>as13000_ipsan_pools = ['Pool0']</code>	列表值	存储池 Cinder 应使用逗号分隔的列表。
<code>as13000_meta_pool = None</code>	字符串值	创建卷时用作元池的池，它应该是存在的复制池。如果没有设置，则驱动程序将从 <code>as13000_ipsan_pools</code> 的值选择一个复制池。
<code>as13000_token_available_time = 3300</code>	整数值	令牌的有效性（以秒为单位）。

配置选项 = 默认值	类型	描述
auth_strategy = keystone	字符串值	用于 auth 的策略。支持 noauth 或 keystone。
az_cache_duration = 3600	整数值	在内存中缓存卷可用区，持续提供的时间（以秒为单位）
backdoor_port = None	字符串值	启用事件程序后端。可接受的值为 0、<port> 和 <start>:<end>，其中 0 会导致侦听随机的 tcp 端口号；<port> 会导致监听指定的端口号（如果正在使用该端口，则不会启用 backdoor）；<start>:<end> 会在指定的端口号范围内侦听最小未使用的端口号。所选端口显示在服务的日志文件中。
backdoor_socket = None	字符串值	启用 eventlet 后端，使用提供的路径作为可接收连接的 unix 套接字。这个选项与中的 <i>backdoor_port</i> 相互排斥，该选项只应提供一个。如果提供这两个选项，则此选项的存在会覆盖该选项的使用。在路径 {pid} 中将被当前进程的 PID 替换。
backend_availability_zone = None	字符串值	此卷后端的可用区。如果没有设置，则 <i>storage_availability_zone</i> 选项值将用作所有后端的默认值。
backend_stats_polling_interval = 60	整数值	后端使用统计请求间隔的时间（以秒为单位）。请注意，对于某些后端生成用量统计非常昂贵，因此设置这个值太低可能会影响性能。
backup_api_class = cinder.backup.api.API	字符串值	卷备份 API 类的完整类名称
backup_ceph_chunk_size = 134217728	整数值	备份在传输到 Ceph 对象存储之前被分成的块大小。
backup_ceph_conf = /etc/ceph/ceph.conf	字符串值	要使用的 Ceph 配置文件。
backup_ceph_image_journals = False	布尔值	如果为 True，将 JOURNALING 和 EXCLUSIVE_LOCK 功能位应用到备份 RBD 对象以允许镜像
backup_ceph_pool = backups	字符串值	存储卷备份的 Ceph 池。
backup_ceph_stripe_count = 0	整数值	创建备份镜像时要使用的 RBD 条带数。
backup_ceph_stripe_unit = 0	整数值	创建备份镜像时要使用的 RBD 条带单元。

配置选项 = 默认值	类型	描述
backup_ceph_user = cinder	字符串值	要连接的 Ceph 用户。在这里，使用与 Cinder 卷相同的用户。如果不使用 cephx，则这应设置为 None。
backup_compression_algorithm = zlib	字符串值	要禁用的压缩算法("none")
backup_container = None	字符串值	用于备份的自定义目录。
backup_driver = cinder.backup.drivers.swift.SwiftBackupDriver	字符串值	用于备份的驱动程序。
backup_driver_init_check_interval = 60	整数值	检查备份驱动程序是否已成功初始化之间的时间（以秒为单位）。
backup_driver_stats_polling_interval = 60	整数值	检查备份驱动程序状态之间的时间（以秒为单位）。如果没有报告工作，则会重启。
backup_enable_progress_timer = True	布尔值	启用或禁用计时器，在将卷备份到后端存储时，将定期进度通知发送到 Ceilometer。默认值为 True，以启用计时器。
backup_file_size = 1999994880	整数值	用于存放备份的文件的最大大小，以字节为单位。如果备份的卷超过这个大小，则它将备份到多个文件中。backup_file_size 必须是 backup_sha_block_size_bytes 的倍数。
backup_manager = cinder.backup.manager.BackupManager	字符串值	卷备份的 Manager 的完整类名称
backup_max_operations = 15	整数值	并发内存重度操作的最大数量：备份和恢复。值 0 表示无限
backup_metadata_version = 2	整数值	备份卷元数据时使用的元数据版本。如果这个数字被禁止，请确保执行恢复的服务支持新版本。
backup_mount_attempts = 3	整数值	在引发错误前挂载 NFS 共享的次数。
backup_mount_options = None	字符串值	传递给 NFS 客户端的挂载选项。详情请查看 NFS 手册页。
backup_mount_point_base = \$state_path/backup_mount	字符串值	包含 NFS 共享的挂载点的基本目录。

配置选项 = 默认值	类型	描述
backup_name_template = backup-%s	字符串值	用于生成备份名称的模板字符串
backup_native_threads_pool_size = 60	整数值	备份的原生线程池的大小。大多数备份驱动都会大量依赖于这个设置，一些特定的驱动则对它的依赖不大。
backup_object_number_per_notification = 10	整数值	发送一个 Ceilometer 通知的块或对象数量
backup_posix_path = \$state_path/backup	字符串值	指定存储备份的位置的路径。
backup_s3_block_size = 32768	整数值	为增量备份跟踪更改的大小（以字节为单位）。 backup_s3_object_size 必须是 backup_s3_block_size 的倍数。
backup_s3_ca_cert_file = None	字符串值	path/to/cert/bundle.pem - 要使用的 CA 证书捆绑包的文件名。
backup_s3_enable_progress_timer = True	布尔值	启用或禁用计时器，在将卷备份到 S3 后端存储时，向 Ceilometer 发送定期进度通知。默认值为 True，以启用计时器。
backup_s3_endpoint_url = None	字符串值	S3 服务器侦听的 url。
<code>`backup_s3_http_proxy = `</code>	字符串值	http 代理服务器的地址或主机。
<code>`backup_s3_https_proxy = `</code>	字符串值	https 代理服务器的地址或主机。
backup_s3_max_pool_connections = 10	整数值	在连接池中保留的最大连接数。
backup_s3_md5_validation = True	布尔值	在 s3 后端中启用或禁用 md5 验证。
backup_s3_object_size = 52428800	整数值	S3 备份对象的大小（以字节为单位）
backup_s3_retry_max_attempts = 4	整数值	代表在单个请求上进行的重试尝试的最大数量的整数。
backup_s3_retry_mode = legacy	字符串值	代表重试模式类型的字符串，例如：legacy、standard、adaptive

配置选项 = 默认值	类型	描述
backup_s3_sse_customer_algorithm = None	字符串值	必须同时设置 SSECustomerAlgorithm. backup_s3_sse_customer_key 来启用 SSE。
backup_s3_sse_customer_key = None	字符串值	SSECustomerKey. backup_s3_sse_customer_algorithm 必须同时设置, 以启用 SSE。
backup_s3_store_access_key = None	字符串值	S3 查询令牌访问密钥。
backup_s3_store_bucket = volumebackups	字符串值	用于存储 Cinder 备份数据的 S3 存储桶。
backup_s3_store_secret_key = None	字符串值	S3 查询令牌 secret 密钥。
backup_s3_timeout = 60	浮点值	抛出超时异常的时间（以秒为单位）。
backup_s3_verify_ssl = True	布尔值	启用或禁用 ssl 验证。
backup_service_inithost_offload = True	布尔值	在备份服务启动过程中卸载待处理的备份删除。如果为 false, 则备份服务将保持关闭, 直到所有待处理的备份都被删除。
backup_sha_block_size_bytes = 32768	整数值	为增量备份跟踪更改的大小（以字节为单位）。 backup_file_size 必须是 backup_sha_block_size_bytes 的倍数。
backup_share = None	字符串值	hostname:path、ipv4addr:path 或 "[ipv6addr]:path" 格式的 NFS 共享。
backup_swift_auth = per_user	字符串值	Swift 身份验证机制 (per_user 或 single_user)。
backup_swift_auth_insecure = False	布尔值	在向 Swift 进行 SSL 连接时绕过服务器证书验证。
backup_swift_auth_url = None	URI 值	Keystone 端点的 URL
backup_swift_auth_version = 1	字符串值	Swift 身份验证版本。为 auth 1.0 指定 "1", 为 auth 2.0 指定 "2" 或 "3" 用于 auth 3.0
backup_swift_block_size = 32768	整数值	为增量备份跟踪更改的大小（以字节为单位）。 backup_swift_object_size 必须是 backup_swift_block_size 的倍数。

配置选项 = 默认值	类型	描述
backup_swift_ca_cert_file = None	字符串值	用于 swift 客户端请求的 CA 证书文件的位置。
backup_swift_container = volumebackups	字符串值	要使用的默认 Swift 容器
backup_swift_enable_progress_timer = True	布尔值	启用或禁用计时器，在将卷备份到 Swift 后端存储时，向 Ceilometer 发送定期进度通知。默认值为 True，以启用计时器。
backup_swift_key = None	字符串值	用于身份验证的 Swift 密钥
backup_swift_object_size = 52428800	整数值	Swift 备份对象的大小（以字节为单位）
backup_swift_project = None	字符串值	Swift 项目/帐户名称。连接到 auth 3.0 系统时需要
backup_swift_project_domain = None	字符串值	Swift 项目域名。连接到 auth 3.0 系统时需要
backup_swift_retry_attempts = 3	整数值	Swift 操作的重试次数
backup_swift_retry_backoff = 2	整数值	Swift 重试之间的 backoff 时间（以秒为单位）
backup_swift_tenant = None	字符串值	Swift 租户/帐户名称。连接到 auth 2.0 系统时需要
backup_swift_url = None	URI 值	Swift 端点的 URL
backup_swift_user = None	字符串值	Swift 用户名
backup_swift_user_domain = None	字符串值	Swift 用户域名。连接到 auth 3.0 系统时需要
backup_timer_interval = 120	整数值	两个进度通知报告备份状态间隔（以秒为单位）
backup_use_same_host = False	布尔值	备份服务使用相同的后端。
backup_use_temp_snaps hot = False	布尔值	如果设置为 True，则会创建一个临时快照来执行非破坏性备份。否则，将克隆临时卷来执行备份。

配置选项 = 默认值	类型	描述
backup_workers = 1	整数值	要启动的备份进程数量。通过并发备份提高性能。
capacity_weight_multiplier = 1.0	浮点值	用于权衡空闲容量的倍数。负数意味着堆栈与分布。
<code>`chap_password = `</code>	字符串值	指定的 CHAP 帐户名称的密码。
chap_password_len = 12	整数值	CHAP 密码的随机字符串的长度。
<code>`chap_username = `</code>	字符串值	CHAP 用户名。
chiscsi_conf = /etc/chelsio-iscsi/chiscsi.conf	字符串值	chiscsi (CXT)全局默认配置文件
cinder_internal_tenant_project_id = None	字符串值	将用作 Cinder 内部租户的项目 ID。
cinder_internal_tenant_user_id = None	字符串值	作为 Cinder 内部租户在卷操作中使用的用户的 ID。
client_socket_timeout = 900	整数值	客户端连接的套接字操作的超时。如果传入的连接闲置了这个秒数，它将被关闭。值 0 表示等待永久。
clone_volume_timeout = 680	整数值	创建克隆卷超时 自 : 14.0.0 *reason:*FusionStorage cinder 驱动程序使用 Restful 方法重构代码，旧的 CLI 模式是 abandon。因此，这些配置项目不再使用。
cloned_volume_same_az = True	布尔值	确保新卷与快照或源卷相同
cluster = None	字符串值	此集群的名称。用于对共享相同后端配置以 HA Active-Active 模式工作的卷主机进行分组。
compression_format = gzip	字符串值	镜像上传时的镜像压缩格式
compute_api_class = cinder.compute.nova.API	字符串值	要使用的计算 API 类的完整类名称

配置选项 = 默认值	类型	描述
config-dir = ['~/project/project.conf.d/ '~/project.conf.d/', '/etc/project/project.conf. d/', '/etc/project.conf.d/']	列表值	从中拉取 conf 文件的配置 目录的路径。此文件集是排序的，例如，如果单个选项被覆盖，则提供可预测的解析顺序。这个集合会在通过前面的 <code>--config-file</code> 指定的文件后解析，因此目录中的超额选项具有优先权。这个选项必须从命令行设置。
config-file = ['~/project/project.conf', '~/project.conf', '/etc/project/project.conf', '/etc/project.conf']	unknown 值	要使用的配置文件的路径。可以指定多个配置文件，后面文件中的值具有优先权。默认为 <code>%(default)s</code> 。这个选项必须从命令行设置。
config_source = []	列表值	列出配置组，它们提供了从本地文件以外的位置访问配置设置的更多详细信息。
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
consistencygroup_api_class = cinder.consistencygroup. api.API	字符串值	consistencygroup API 类的完整类名称
control_exchange = openstack	字符串值	默认交换，哪些主题在哪些方面有范围。可以被 <code>transport_url</code> 选项指定的交换名称覆盖。
datara_503_interval = 5	整数值	503 重试之间的间隔
datara_503_timeout = 120	整数值	HTTP 503 重试消息的超时时间
datara_api_port = 7717	字符串值	Datera API 端口。
datara_api_version = 2.2	字符串值	Datera API 版本。
datara_debug = False	布尔值	true 设置功能 <code>arg</code> 和返回日志记录
datara_debug_replica_count_override = False	布尔值	仅 FOR DEBUG/TESTING PURPOSES True 将 <code>replica_count</code> 设置为 1
datara_disable_extended_metadata = False	布尔值	设置为 True 以禁用向 Datera 后端发送额外元数据
datara_disable_profiler = False	布尔值	设置为 True 以禁用 Datera 驱动程序中的性能分析

配置选项 = 默认值	类型	描述
datara_disable_template_override = False	布尔值	设置为 True，以在从模板创建时禁用 size 属性的自动模板覆盖
datara_enable_image_cache = False	布尔值	设置为 True 以启用 Datera 后端镜像缓存
datara_image_cache_volume_type_id = None	字符串值	用于缓存卷的 Cinder 卷类型 id
datara_ldap_server = None	字符串值	LDAP 身份验证服务器
datara_tenant_id = None	字符串值	如果设置为 Map --> OpenStack 项目 ID 将隐式映射到 Datera 租户 ID，如果设置为 None --> Datera 租户 ID 不会在卷置备过程中使用（如果设置为任何其他内容 --> Datera 租户 ID）将提供的值
datara_volume_type_defaults = {}	dict 值	如果没有提供 volume-type 设置，则此处的设置将用作卷类型默认值。例如，这可用于设置非常低的 total_iops_max 值，如果在 volume-type 中指定 none，以防止意外过度使用。选项通过以下格式指定，WITHOUT ANY DF: PREFIX: <i>datara_volume_type_defaults=iops_per_gb:100,bandwidth_per_gb:200...etc.</i>
db_driver = cinder.db	字符串值	用于数据库访问的驱动
debug = False	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。
default_availability_zone = None	字符串值	新卷的默认可用区。如果没有设置，则 storage_availability_zone 选项值将用作新卷的默认值。
default_group_type = None	字符串值	要使用的默认组类型

配置选项 = 默认值	类型	描述
default_log_levels = ['amqp=WARN', 'amqplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。 如果设置了 log_config_append，则忽略此选项。
default_sandstone_target_ips = []	列表值	SandStone 默认目标 ip.
default_volume_type = __DEFAULT__	字符串值	要使用的默认卷类型
driver_client_cert = None	字符串值	如果驱动程序支持客户端证书的路径，进行验证。
driver_client_cert_key = None	字符串值	如果驱动程序支持客户端证书密钥的路径，进行验证。
driver_data_namespace = None	字符串值	要保存到的驱动程序私有数据值的命名空间。
driver_ssl_cert_path = None	字符串值	可以用来指定到带有可信 CA 证书的 CA_BUNDLE 文件或目录的非默认路径，它将用于验证后端
driver_ssl_cert_verify = False	布尔值	如果设置为 True，http 客户端将验证后端端点的 SSL 证书。

配置选项 = 默认值	类型	描述
driver_use_ssl = False	布尔值	如果驱动程序支持，告诉驱动程序使用 SSL 连接到后端存储。
dsware_isthin = False	布尔值	精简存储分配的标记。 弃用自 : 14.0.0 *reason:*FusionStorage cinder 驱动程序使用 Restful 方法重构代码，旧的 CLI 模式是 abandon。因此，这些配置项目不再使用。
<code>`dsware_manager = `</code>	字符串值	cinder-volume 的 Fusionstorage 管理器 ip addr. 弃用自 : 14.0.0 *reason:*FusionStorage cinder 驱动程序使用 Restful 方法重构代码，旧的 CLI 模式是 abandon。因此，这些配置项目不再使用。
<code>`dsware_rest_url = `</code>	字符串值	FusionStorage 数组的地址。例如，"dsware_rest_url=xxx"
<code>`dsware_storage_pools = `</code>	字符串值	FusionStorage 数组上的池列表，分号(;)用于分割存储池，"dsware_storage_pools = xxx1; xxx2; xxx3"
enable_force_upload = False	布尔值	在 upload_to_image 上启用 Force 选项。这启用了在正在使用的卷上运行 upload_volume，用于支持它的后端。
enable_new_services = True	布尔值	要创建时添加到可用池中的服务
enable_unsupported_driver = False	布尔值	当您允许不支持的驱动程序启动时，将其设置为 True。没有维护正常工作的 CI 系统和测试的驱动程序被标记为不受支持，直到 CI 再次工作为止。这也会将驱动程序标记为已弃用，并可在下一发行版本中删除。
enable_v2_api = True	布尔值	DEPRECATED : 部署 Cinder API 的 v2。
enable_v3_api = True	布尔值	部署 Cinder API 的 v3。
enabled_backends = None	列表值	要使用的后端名称列表。这些后端名称应由一个唯一的 [CONFIG] 组支持，及其选项
enforce_multipath_for_image_xfer = False	布尔值	如果设置为 True，则当 multipathd 未运行时，镜像传输的卷附加将中止。否则，它将回退到单一路径。这个参数需要为每个 backend 部分或 [backend_defaults] 部分中配置，作为所有后端的通用配置。

配置选项 = 默认值	类型	描述
executor_thread_pool_size = 64	整数值	当 executor 为 threading 或 eventlet 时，executor 线程池的大小。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
filter_function = None	字符串值	将用于过滤主机的模拟字符串。仅在将驱动程序过滤器设置为供 Cinder 调度程序使用时使用。
<code>fusionstorageagent = `</code>	字符串值	Fusionstorage 代理 ip addr 范围 已弃用 : 14.0.0 *reason:*FusionStorage cinder 驱动程序使用 Restful 方法重构代码，旧的 CLI 模式是 abandon。因此，这些配置项目不再使用。
glance_api_insecure = False	布尔值	允许执行对 glance 的不安全 SSL (https) 请求 (将使用 https，但不会执行证书验证)。
glance_api_servers = None	列表值	可用于 cinder 的 glance API 服务器的 URL 列表 ([http[s]://][hostname ip]:port)。如果没有指定协议，则默认为 http。
glance_api_ssl_compression = False	布尔值	启用或禁用 SSL 层压缩的协商。在某些情况下，禁用压缩可以提高数据吞吐量，比如当高网络带宽可用且使用 qcow2 等压缩镜像格式时。
glance_ca_certificates_file = None	字符串值	用于 glance 客户端请求的 ca 证书文件的位置。
glance_catalog_info = image:glance:publicURL	字符串值	在服务目录中查找 glance 时匹配的信息。格式是：分隔的值，格式为：<service_type><service_name>:<endpoint_type> - 只有在未提供 glance_api_servers 时才使用。
glance_certfile = None	字符串值	用于 glance 客户端请求的证书文件的位置。
glance_core_properties = ['checksum', 'container_format', 'disk_format', 'image_name', 'image_id', 'min_disk', 'min_ram', 'name', 'size']	列表值	镜像的默认内核属性
glance_keyfile = None	字符串值	用于 glance 客户端请求的证书密钥文件的位置。
glance_num_retries = 3	整数值	从 glance 下载镜像时的重试次数

配置选项 = 默认值	类型	描述
glance_request_timeout = None	整数值	glance 操作的 HTTP/https 超时值。此处不提供值 (None)，则使用 glanceclient 默认值。
glusterfs_backup_mount_point = \$state_path/backup_mount	字符串值	包含 gluster 共享的挂载点的基本目录。
glusterfs_backup_share = None	字符串值	<hostname ipv4addr ipv6addr>:{r_vol_nam} 格式的 GlusterFS 共享。eg: 1.2.3.4:backup_vol
goodness_function = None	字符串值	用于确定主机的良好性的模拟字符串。仅在使用良好性 weigher 时使用，供 Cinder 调度程序使用。
graceful_shutdown_timeout = 60	整数值	指定安全关闭服务器将退出的超时。零值表示无限期等待。
group_api_class = cinder.group.api.API	字符串值	组 API 类的完整类名称
hitachi_mirror_auth_password = None	字符串值	iSCSI 验证密码
hitachi_mirror_auth_user = None	字符串值	iSCSI 身份验证用户名
hitachi_mirror_compute_target_ports = []	列表值	主机组或 iSCSI 目标的目标节点端口名称
hitachi_mirror_ldev_range = None	字符串值	辅助存储系统的逻辑设备范围
hitachi_mirror_pair_target_number = 0	整数值	主机组或 iSCSI 目标的目标名称
hitachi_mirror_pool = None	字符串值	辅助存储系统的池
hitachi_mirror_rest_api_ip = None	字符串值	REST API 服务器的 IP 地址
hitachi_mirror_rest_api_port = 443	端口值	REST API 服务器的端口号

配置选项 = 默认值	类型	描述
hitachi_mirror_rest_pair_target_ports = []	列表值	主机组或 iSCSI 目标对的目标端口名称
hitachi_mirror_rest_password = None	字符串值	REST API 的辅助存储系统的密码
hitachi_mirror_rest_user = None	字符串值	REST API 的辅助存储系统的用户名
hitachi_mirror_snap_pool = None	字符串值	辅助存储系统的精简池
hitachi_mirror_ssl_cert_path = None	字符串值	可以用来指定到带有可信 CA 证书的 CA_BUNDLE 文件或目录的非默认路径，它将用于验证后端
hitachi_mirror_ssl_cert_verify = False	布尔值	如果设置为 True，http 客户端将验证后端端点的 SSL 证书。
hitachi_mirror_storage_id = None	字符串值	辅助存储系统的 ID
hitachi_mirror_target_ports = []	列表值	主机组或 iSCSI 目标的目标端口名称
hitachi_mirror_use_chap_auth = False	布尔值	是否要使用 iSCSI 身份验证
hitachi_path_group_id = 0	整数值	分配给远程复制的远程连接的路径组 ID
hitachi_quorum_disk_id = None	整数值	用于全局主动设备的 Quorum 磁盘的 ID
hitachi_replication_copy_speed = 3	整数值	存储系统的远程副本速度。1 或 2 表示低速度，3 表示中速度，4 到 15 之间的值表示高速度。
hitachi_replication_number = 0	整数值	REST API 的实例号
hitachi_replication_status_check_long_interval = 600	整数值	检查远程复制对状态的时间间隔。如果在此参数指示的时间过后，状态没有更改为预期状态，则应用此参数。
hitachi_replication_status_check_short_interval = 5	整数值	检查远程复制对状态的初始间隔
hitachi_replication_status_check_timeout = 86400	整数值	远程复制对状态更改为预期状态前的最大等待时间

配置选项 = 默认值	类型	描述
hitachi_set_mirror_reserve_attribute = True	布尔值	是否设置镜像保留属性
Host = <based on operating system>	字符串值	此节点的名称。这可以是不透明标识符。它不一定是主机名、FQDN 或 IP 地址。
iet_conf = /etc/iet/ietd.conf	字符串值	DEPRECATED: IET 配置文件
image_compress_on_upload = True	布尔值	在可能的情况下，压缩上传到镜像服务的镜像
image_conversion_address_space_limit = 1	整数值	以 GB 为单位的地址空间限制来转换镜像
image_conversion_cpu_limit = 60	整数值	转换镜像的 CPU 时间限制（以秒为单位）
image_conversion_dir = \$state_path/conversion	字符串值	在镜像转换过程中用于临时存储的目录
image_upload_use_cinder_backend = False	布尔值	如果设置为 True，则以原始格式的 upload-to-image 将创建一个克隆的卷，并将其位置注册到镜像服务，而不是上传卷内容。在镜像服务中必须启用 cinder 后端和位置支持。
image_upload_use_internal_tenant = False	布尔值	如果设置为 True，则由 upload-to-image 创建的镜像卷将放入内部租户中。否则，镜像卷会在当前上下文的租户中创建。
image_volume_cache_enabled = False	布尔值	为此后端启用镜像卷缓存。
image_volume_cache_max_count = 0	整数值	镜像卷缓存中允许的最大条目数。0 jpeg unlimited。
image_volume_cache_max_size_gb = 0	整数值	此后端的镜像卷缓存的最大大小（以 GB 为单位）。0 jpeg unlimited。
infortrend_cli_cache = False	布尔值	Infortrend CLI 缓存。当设置 True 时，RAID 状态报告将使用存储在 CLI 中的缓存。除非 RAID 仅由 Openstack 管理，且仅由 infortrend cinder-volume 后端管理，否则请不要启用此功能。否则，CLI 可能会向 cinder 报告过时的状态，因此所有后端/CLI 中可能存在一些竞争条件。
infortrend_cli_max_retries = 5	整数值	如果命令失败，则最大重试次数。

配置选项 = 默认值	类型	描述
infortrend_cli_path = /opt/bin/Infortrend/raidcmd_ESDS10.jar	字符串值	Infortrend CLI 绝对路径。
infortrend_cli_timeout = 60	整数值	CLI 的超时时间（以秒为单位）。
infortrend_iqn_prefix = iqn.2002-10.com.infortrend	字符串值	iSCSI 的 Infortrend iqn 前缀。
<code>`infortrend_pools_name = `</code>	列表值	Infortrend 逻辑卷名称列表。它用逗号分开。
<code>`infortrend_slots_a_channels_id = `</code>	列表值	用于 OpenStack 的 Slot A 上的 Infortrend raid 频道 ID 列表。它用逗号分开。
<code>`infortrend_slots_b_channels_id = `</code>	列表值	用于 OpenStack 的 Slot B 上的 Infortrend raid 频道 ID 列表。它用逗号分开。
init_host_max_objects_retrieval = 0	整数值	在卷管理器主机初始化期间每个批处理检索的卷和快照的最大数量。查询结果将从数据库批量获得，而不是在一个 shot 中获取，以避免极端内存用量。设置 0 以关闭此功能。
initiator_assign_sandstone_target_ip = {}	dict 值	支持启动器分配目标并分配 ip。
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。
instorage_mcs_allow_tenant_qos = False	布尔值	允许租户在创建时指定 QOS
instorage_mcs_iscsi_chap_enabled = True	布尔值	为 iSCSI 连接配置 CHAP 身份验证（默认：Enabled）
instorage_mcs_localcopy_rate = 50	整数值	指定创建完整卷副本时要使用的 InStorage LocalCopy 复制率。默认率为 50，有效率为 1-100。
instorage_mcs_localcopy_timeout = 120	整数值	等待 LocalCopy 准备的最大秒数。

配置选项 = 默认值	类型	描述
instorage_mcs_vol_autoexpand = True	布尔值	卷的存储系统 autoexpand 参数(True/False)
instorage_mcs_vol_compression = False	布尔值	卷的存储系统压缩选项
instorage_mcs_vol_grain_size = 256	整数值	存储系统获取卷的大小参数(32/64/128/256)
instorage_mcs_vol_intier = True	布尔值	为卷启用 InTier
instorage_mcs_vol_iogrp = 0	字符串值	分配卷的 I/O 组。它可以是一个用逗号分隔的列表，在这种情况下，驱动程序将根据与 io_group 关联的最少卷数量选择 io_group。
instorage_mcs_vol_rsize = 2	整数值	用于卷的存储系统空间效率参数（百分比）
instorage_mcs_vol_warning = 0	整数值	卷容量警告的存储系统阈值（百分比）
instorage_mcs_volpool_name = ['volpool']	列表值	卷存储系统存储池的逗号分隔列表。
instorage_san_secondary_ip = None	字符串值	指定在 san_ip 无效或无法访问时使用的辅助管理 IP 或主机名。
iscsi_iotype = fileio	字符串值	将 iSCSI 目标的行为设置为执行 blockio 或 fileio，可以设置 auto，Cinder 将自动检测后备设备的类型
<code>`iscsi_target_flags = `</code>	字符串值	为 iSCSI 目标设置特定于目标的标记。仅用于 tgtadm 来使用 bsoflags 选项指定后备设备标志。指定的字符串被传递为底层工具。
iscsi_write_cache = on	字符串值	将 iSCSI 目标的行为设置为执行回写(on)或 write-through (off)。如果 target_helper 设置为 tgtadm，则此参数有效。
iser_helper = tgtadm	字符串值	要使用的 iSER 目标用户固定工具的名称
iser_ip_address = \$my_ip	字符串值	iSER 守护进程侦听的 IP 地址
iser_port = 3260	端口值	iSER 守护进程侦听的端口

配置选项 = 默认值	类型	描述
iser_target_prefix = iqn.2010- 10.org.openstack:	字符串值	iSER 卷的前缀
java_path = /usr/bin/java	字符串值	Java 绝对路径。
jovian_block_size = 64K	字符串值	块大小可以是：32K、64K、128K、256K、512K、1M
jovian_ignore_tpath = []	列表值	要忽略的多路径 IP 地址列表。
jovian_pool = Pool-0	字符串值	保存所有 cinder 卷的 JovianDSS 池
jovian_recovery_delay = 60	整数值	HA 集群失败前的时间。
keystone_catalog_info = identity:Identity Service:publicURL	字符串值	在服务目录中查找 keystone 时匹配的信息。格式是：分隔的格式为：<service_type>:<service_name>:<endpoint_type> - 仅当 backup_swift_auth_url 未设置时才使用
kioxia_block_size = 4096	整数值	卷块大小（以字节为单位） - 512 或 4096（默认）。
kioxia_cafile = None	字符串值	provisioner REST API SSL 的证书
kioxia_desired_bw_per_g b = 0	整数值	每个 GB 的 B/s 所需的带宽。
kioxia_desired_iops_per_ gb = 0	整数值	所需的 IOPS/GB。
kioxia_max_bw_per_gb = 0	整数值	每个 GB 的 B/s 中带宽的上限。
kioxia_max_iops_per_gb = 0	整数值	IOPS/GB 的上限。
kioxia_max_replica_down _time = 0	整数值	复制的卷最大停机时间（以分钟为单位）。
kioxia_num_replicas = 1	整数值	卷副本数。
kioxia_provisioning_type = THICK	字符串值	精简卷或厚卷，默认厚。

配置选项 = 默认值	类型	描述
kioxia_same_rack_allowed = False	布尔值	多个副本可以分配给同一机架。
kioxia_snap_reserved_space_percentage = 0	整数值	用于日志的父卷的百分比。
kioxia_snap_vol_reserved_space_percentage = 0	整数值	用于日志的父卷的可写快照百分比。
kioxia_snap_vol_span_allowed = True	布尔值	Allow span in snapshot volume - Default True。
kioxia_span_allowed = True	布尔值	Allow span - Default True。
kioxia_token = None	字符串值	KumoScale Provisioner 身份验证令牌。
kioxia_url = None	字符串值	KumoScale provisioner REST API URL
kioxia_vol_reserved_space_percentage = 0	整数值	精简卷保留容量分配百分比。
kioxia_writable = False	布尔值	来自快照写入的卷。
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 log-date-format）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_options = True	布尔值	启动服务时，启用或禁用所有注册选项的日志值（在 DEBUG 级别）。

配置选项 = 默认值	类型	描述
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval", 否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时, 将使用上次文件更改的时间 (或服务启动的时间)。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 DEBUG 时, 要附加到日志消息的其他数据。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_default_format_s tring = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_user_identity_for mat = %(user) s %(tenant) s %(domain) s % (user_domain) s % (project_domain) s	字符串值	定义 <code>logging_context_format_string</code> 中使用的 <code>%(user_identity) s</code> 的格式字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
manager_ips = {}	dict 值	这个选项是支持 FSA 在不同节点间挂载。这些参数采用标准字典配置表单 <code>manager_ips = host1:ip1, host2:ip2...</code>
max_age = 0	整数值	后续使用刷新之间的秒数
max_header_line = 16384	整数值	要接受的最大消息标头大小。在使用大令牌时可能需要增加 <code>max_header_line</code> (通常是在 keystone 配置为使用具有大型服务目录的 PKI 令牌时生成的)。

配置选项 = 默认值	类型	描述
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
max_over_subscription_ratio = 20.0	字符串值	启用精简配置时，超订阅比率的表示。默认比率为 20.0，这意味着置备的容量可以是总物理容量的 20 倍。如果比率是 10.5，这表示调配的容量可以是总物理容量的 10.5 倍。比率 1.0 表示调配的容量不能超过物理容量的总容量。如果比率是 <i>auto</i> ，Cinder 将根据调配的容量和已用空间自动计算比率。如果没有设置 <i>auto</i> ，则比率必须至少为 1.0。
message_reap_interval = 86400	整数值	定期任务运行之间的间隔，以清理过期的消息（以秒为单位）。
message_ttl = 2592000	整数值	消息最小生命周期（以秒为单位）。
migration_create_volume_timeout_secs = 300	整数值	创建卷时迁移到的超时（秒）
monkey_patch = False	布尔值	启用 monkey 补丁
monkey_patch_modules = []	列表值	模块/decorators 到 monkey 补丁列表
my_ip = <based on operating system>	主机地址值	此主机的 IP 地址
no_snapshot_gb_quota = False	布尔值	是针对 GB 配额的快照数
num_iser_scan_tries = 3	整数值	重新扫描 iSER 目标查找卷的次数上限
num_shell_tries = 3	整数值	尝试运行 flakey shell 命令的次数
num_volume_device_scan_tries = 3	整数值	重新扫描目标以查找卷的次数上限
nvmeof_conn_info_version = 1	整数值	NVMe os-brick 连接器有 2 种不同的连接信息格式，这允许一些使用原始格式（版本 1）的 NVMe-oF 驱动程序（如 spdk 和 LVM-nvmet）来发送更新的格式。
nvmet_ns_id = 10	整数值	没有共享目标时 LVM 卷的子系统的命名空间 ID。共享时的最小 id 值。Linux 中支持的最大值为 8192

配置选项 = 默认值	类型	描述
nvmet_port_id = 1	端口值	没有共享目标时 NVMe 目标端口定义的 id。共享时起始端口 id 值，为每个二级 ip 地址递增。
osapi_max_limit = 1000	整数值	集合资源在单个响应中返回的最大项目数
osapi_volume_ext_list = []	列表值	在 cinder.api.contrib.select_extensions 中使用 osapi_volume_extension 选项时，指定载入的扩展列表
osapi_volume_extension = ['cinder.api.contrib.standard_extensions']	多值	要载入的 osapi 卷扩展
osapi_volume_listen = 0.0.0.0	字符串值	OpenStack Volume API 侦听的 IP 地址
osapi_volume_listen_port = 8776	端口值	OpenStack Volume API 侦听的端口
osapi_volume_use_ssl = False	布尔值	如果设置了 True，请将套接字嵌套在 SSL 上下文中。必须指定证书文件和密钥文件。
osapi_volume_workers = None	整数值	OpenStack Volume API 服务的 worker 数量。默认值等于可用的 CPU 数量。
per_volume_size_limit = -1	整数值	每个卷允许的最大大小，以 GB 为单位
periodic_fuzzy_delay = 60	整数值	在启动定期任务调度程序时，范围（以秒为单位）会随机延迟，以减少时间戳。（通过设置为 0 来禁用）
periodic_interval = 60	整数值	运行定期任务之间的间隔（以秒为单位）
pool_id_filter = []	列表值	池 id 允许使用 已弃用，因为：14.0.0 *reason:*FusionStorage cinder 驱动程序使用 Restful 方法重构代码，旧的 CLI 模式是 abandon。因此，这些配置项目不再使用。
pool_type = default	字符串值	池类型，如 sata-2copy 弃用，自：14.0.0 *reason:*FusionStorage cinder 驱动程序使用 Restful 方法重构代码，旧的 CLI 模式是 abandon。因此，这些配置项目不再使用。

配置选项 = 默认值	类型	描述
public_endpoint = None	字符串值	用于版本端点的公共 url。默认值为 None，它将使用请求的 host_url 属性来填充 URL 基础。如果 Cinder 在代理后面运行，则需要更改它来代表代理的 URL。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
quota_backup_gigabytes = 1000	整数值	每个项目允许备份存储总量（以 GB 为单位）
quota_backups = 10	整数值	每个项目允许的卷备份数
quota_consistencygroups = 10	整数值	每个项目允许的一致性组数
quota_driver = cinder.quota.DbQuotaDriver	字符串值	用于配额检查的默认驱动程序
quota_gigabytes = 1000	整数值	每个项目允许的卷和快照总量（以 GB 为单位）
quota_groups = 10	整数值	每个项目允许的组数
quota_snapshots = 10	整数值	每个项目允许的卷快照数
quota_volumes = 10	整数值	每个项目允许的卷数
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
reinit_driver_count = 3	整数值	如果卷初始化失败，则调整驱动程序的最大次数。重试的时间间隔是指数避退的 backoff，将是 1s、2s、4s 等。
replication_device = None	dict 值	多选择代表复制目标设备的字典。这个选项可以在单一配置部分中多次指定，以指定多个复制目标设备。每个条目都使用标准字典配置表单： replication_device = target_device_id: <required>,key1:value1,key2:value2...

配置选项 = 默认值	类型	描述
report_discard_supported = False	布尔值	报告后端支持丢弃的 Cinder 客户端（也称为 trim/unmap）。这不会实际更改后端或客户端的行为，它只通知它可以被使用。
report_interval = 10	整数值	在节点向数据存储报告状态之间的间隔（以秒为单位）
reservation_clean_interval = \$reservation_expire	整数值	定期任务运行之间的间隔，以清理过期的保留（以秒为单位）。
reservation_expire = 86400	整数值	保留过期前的秒数
reserved_percentage = 0	整数值	后端容量的百分比被保留
resource_query_filters_file = /etc/cinder/resource_filters.json	字符串值	指示用户可见过滤器参数的 JSON 文件用于列表查询。
restore_discard_excess_bytes = True	布尔值	如果为 True，在恢复卷时始终丢弃过量字节数，例如，带有零的 pad。
rootwrap_config = /etc/cinder/rootwrap.conf	字符串值	用于以 root 用户身份运行命令的 rootwrap 配置文件的路径
rpc_conn_pool_size = 30	整数值	RPC 连接池的大小。
rpc_ping_enabled = False	布尔值	添加端点以回答 ping 调用。端点名为 oslo_rpc_server_ping
rpc_response_timeout = 60	整数值	从调用等待响应的秒数。
run_external_periodic_tasks = True	布尔值	某些定期任务可以在单独的进程中运行。我们是否应该在此处运行？
san_hosts = `	列表值	Open-E JovianDSS SA 的 IP 地址
sandstone_pool = `	字符串值	SandStone 存储池资源名称。
scheduler_default_filters = ['AvailabilityZoneFilter', 'CapacityFilter', 'CapabilitiesFilter']	列表值	在请求中未指定时用于过滤主机的类名称。

配置选项 = 默认值	类型	描述
scheduler_default_weighters = ['CapacityWeigher']	列表值	用于权衡主机的类名称。
scheduler_driver = cinder.scheduler.filter_scheduler.FilterScheduler	字符串值	要使用的默认调度程序驱动程序
scheduler_driver_init_wait_time = 60	整数值	等待驱动程序报告为就绪的最大时间（以秒为单位）
scheduler_host_manager = cinder.scheduler.host_manager.HostManager	字符串值	要使用的调度程序主机管理器类
<code>`scheduler_json_config_location = `</code>	字符串值	调度程序配置 JSON 文件的绝对路径。
scheduler_manager = cinder.scheduler.manager.SchedulerManager	字符串值	调度程序的 Manager 的完整类名称
scheduler_max_attempts = 3	整数值	调度卷的最大尝试次数
scheduler_weight_handler = cinder.scheduler.weights.OrderedHostWeightHandler	字符串值	用于在权衡后选择 host/pool 的处理程序
scst_target_driver = iscsi	字符串值	SCST 目标实施可以从多个 SCST 目标驱动程序中选择。
scst_target_iqn_name = None	字符串值	某些 ISCSI 目标具有预定义的目标名称，SCST 目标驱动程序使用此名称。
service_down_time = 60	整数值	最后一次检查后的最大时间，用于考虑服务
snapshot_name_template = snapshot-%s	字符串值	用于生成快照名称的模板字符串
snapshot_same_host = True	布尔值	在快照所在的主机上从快照创建卷
split_loggers = False	布尔值	将请求记录到多个日志记录器。

配置选项 = 默认值	类型	描述
ssh_hosts_key_file = \$state_path/ssh_known_hosts	字符串值	包含 Cinder 需要与之通信的系统的 SSH 主机密钥的文件。OPTIONAL: Default=\$state_path/ssh_known_hosts
state_path = /var/lib/cinder	字符串值	维护 cinder 状态的顶级目录
storage_availability_zone = nova	字符串值	此节点的可用区。可以使用选项 "backend_availability_zone" 覆盖每个卷后端。
storage_protocol = iscsi	字符串值	在主机和存储后端之间传输数据的协议。
strict_ssh_host_key_policy = False	布尔值	启用严格的主机密钥检查的选项。当设置为 "True" 时, Cinder 将仅与配置的 "ssh_hosts_key_file" 中存在的主机密钥连接系统。当设置为 "False" 时, 主机密钥将在第一个连接时保存, 用于后续连接。 Default=False
swift_catalog_info = object-store:swift:publicURL	字符串值	在服务目录中查找 swift 时匹配的信息。格式是: 以逗号分隔的值, 格式为: <service_type>: <service_name>:<endpoint_type> - 仅当 backup_swift_url 未设置时才使用
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append, 则忽略此选项。
target_helper = tgtadm	字符串值	使用目标用户固定工具。tgtadm 为默认, 使用 lioadm 作为 LIO iSCSI 支持、SCST 目标支持的 scstadmin、ietadm for iSCSI Enterprise Target、iscsictl 用于 Chelsio iSCSI 目标, nvmet 用于 NVMeoF 支持, spdk-nvmeof 用于 SPDK NVMe-oF, 或为测试虚拟虚拟。注: IET 驱动程序已弃用, 并将在 V 发行版本中删除。
target_ip_address = \$my_ip	字符串值	iSCSI/NVMeoF 守护进程侦听的 IP 地址
target_port = 3260	端口值	iSCSI/NVMeoF 守护进程侦听的端口
target_prefix = iqn.2010-10.org.openstack:	字符串值	iSCSI/NVMeoF 卷的前缀
target_protocol = iscsi	字符串值	决定新卷的目标协议, 使用 tgtadm、lioadm 和 nvmet 目标帮助程序创建。要启用 RDMA, 此参数应该使用值 "iser" 设置。支持的 iSCSI 协议值为 "iscsi" 和 "iser", 如果 nvmet target 设置为 "nvmet_rdma" 或 "nvmet_tcp"。

配置选项 = 默认值	类型	描述
target_secondary_ip_addresses = []	列表值	iSCSI/NVMeoF 守护进程的辅助 IP 地址列表
tcp_keepalive = True	布尔值	为每个服务器套接字设置 TCP_KEEPALIVE (True/False) 的值。
tcp_keepalive_count = None	整数值	为每个服务器套接字设置 TCP_KEEPCNT 的值。在 OS X 上不支持。
tcp_keepalive_interval = None	整数值	设置每个服务器套接字的 TCP_KEEPINTVL 的值（以秒为单位）。在 OS X 上不支持。
tcp_keepidle = 600	整数值	为每个服务器套接字设置 TCP_KEEPIDLE 的值（以秒为单位）。在 OS X 上不支持。
trace_flags = None	列表值	控制将哪些 trace info 写入 DEBUG 日志级别的选项列表，以帮助开发人员。有效值为 method 和 api。
transfer_api_class = cinder.transfer.api.API	字符串值	卷传输 API 类的完整类名称
transport_url = rabbit://	字符串值	<p>用于连接到消息传递后端的网络地址和可选用户凭证，采用 URL 格式。预期格式为：</p> <p>driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query</p> <p>示例：rabbit://rabbitmq:password@127.0.0.1:5672//</p> <p>有关 URL 中字段的详情，请查看 oslo_messaging.TransportURL 文档</p>
until_refresh = 0	整数值	在刷新使用前的保留计数
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。

配置选项 = 默认值	类型	描述
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_chap_auth = False	布尔值	为目标启用/禁用 CHAP 验证的选项。
use_default_quota_class = True	布尔值	启用或禁用将默认配额类与默认配额一起使用。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_forwarded_for = False	布尔值	将 X-Forwarded-For 视为规范远程地址。只有在您有一个 sanitizing 代理时才启用。
use_multipath_for_image_xfer = False	布尔值	我们是否使用多路径(image)将多路径附加到卷并镜像到卷传输中，在 cinder 中附加/分离卷？这个参数需要为每个 backend 部分或 [backend_defaults] 部分中配置，作为所有后端的通用配置。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。
verify_glance_signatures = enabled	字符串值	<p>启用镜像签名验证。</p> <p>Cinder 使用来自 Glance 的镜像签名元数据，并在下载该镜像时验证签名镜像的签名。这里有两个选项。</p> <ol style="list-style-type: none"> 1. 启用 : 验证镜像是否有签名元数据。 2. disabled : 验证关闭。 <p>如果无法验证镜像签名，或者在需要时不完整镜像签名元数据，则 Cinder 不会创建卷并将其更新为错误状态。这为最终用户提供了更强的保证镜像数据的完整性。</p>
vmdk_allowed_types = ['streamOptimized', 'monolithicSparse']	列表值	描述允许 VMDK createType 子格式的字符串列表。我们建议您仅包含 single-file-with-sparse-header 变体，以避免当镜像转换为原始格式时，避免潜在的主机文件暴露。如果此列表为空，则不允许使用 VMDK 镜像。
volume_api_class = cinder.volume.api.API	字符串值	要使用的卷 API 类的完整类名称

配置选项 = 默认值	类型	描述
volume_backend_name = None	字符串值	给定驱动程序实现的后端名称
volume_clear = zero	字符串值	用于擦除旧卷的方法
volume_clear_ionice = None	字符串值	标记用于更改在删除后用于 0 个卷的进程的 i/o 优先级，例如 "-c3" 只用于闲置优先级。
volume_clear_size = 0	整数值	在旧卷开始时擦除大小（以 MiB 为单位）。1024 MiB 为 max. 0 especially all
volume_copy_blkio_cgroup_name = cinder-volume-copy	字符串值	用于限制卷复制带宽的 blkio cgroup 名称
volume_copy_bps_limit = 0	整数值	卷副本的带宽上限。0 jpeg unlimited
volume_dd_blocksize = 1M	字符串值	复制/清理卷时使用的默认块大小
volume_manager = cinder.volume.manager.VolumeManager	字符串值	卷管理器的完整类名称
volume_name_template = volume-%s	字符串值	用于生成卷名称的模板字符串
volume_number_multiplier = -1.0	浮点值	用于权衡卷号的倍数。负数意味着分散与堆栈。
volume_service_inithost_offload = False	布尔值	在卷服务启动过程中卸载待处理的卷删除
volume_transfer_key_length = 16	整数值	自动生成的 auth 密钥中的字符数。
volume_transfer_salt_length = 8	整数值	salt 中的字符数。
volume_usage_audit_period = month	字符串值	生成卷用量的时间周期。这些选项包括小时、天、月份或年。
volumes_dir = \$state_path/volumes	字符串值	卷配置文件存储目录
vrts_lun_sparse = True	布尔值	创建稀疏 Lun。

配置选项 = 默认值	类型	描述
vrts_target_config = /etc/cinder/vrts_target.xml	字符串值	VA 配置文件。
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 <code>log_file</code> 选项并使用 Linux 平台时，才有意义。如果设置了 <code>log_config_append</code> ，则忽略此选项。
wsgi_default_pool_size = 100	整数值	wsgi 使用的 greenthreads 池的大小
wsgi_keep_alive = True	布尔值	如果为 False，请明确关闭客户端套接字连接。
wsgi_log_format = %(client_ip) s "%(request_line) s" status: %(status_code) s len: %(body_length) s time: %(wall_seconds).7f	字符串值	用作模板用于生成日志行的 python 格式字符串。以下值可以格式化到其中： <code>client_ip</code> , <code>date_time</code> , <code>request_line</code> , <code>status_code</code> , <code>body_length</code> , <code>wall_seconds</code> 。
wsgi_server_debug = False	布尔值	如果服务器应该将异常回溯发送到客户端 500 错误，则为 true。如果为 False，服务器将响应空正文。
zoning_mode = None	字符串值	配置了 FC Zoning 模式，现在只支持 <i>fabric</i> 。

2.1.2. 后端

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[backend]` 组下可用的选项。

表 2.1. 后端

配置选项 = 默认值	类型	Description
backend_host = None	字符串值	主机值的后端覆盖。

2.1.3. backend_defaults

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[backend_defaults]` 组下可用的选项。

表 2.2. backend_defaults

配置选项 = 默认值	类型	Description
auto_calc_max_oversubscription_ratio = False	布尔值	K2 驱动程序将在将这个选项设置为 True 时计算 <code>max_oversubscription_ratio</code> 。

配置选项 = 默认值	类型	Description
backend_availability_zone = None	字符串值	此卷后端的可用区。如果没有设置，则 storage_availability_zone 选项值将用作所有后端的默认值。
backend_native_threads_pool_size = 20	整数值	后端的原生线程池的大小。增加大量依赖的后端，如 RBD 驱动程序。
CHAP = disabled	字符串值	CHAP 验证模式，仅对 iscsi 有效（禁用 启用）
<code>`chap_password = `</code>	字符串值	指定的 CHAP 帐户名称的密码。
<code>`chap_username = `</code>	字符串值	CHAP 用户名。
check_max_pool_luns_threshold = False	布尔值	DEPRECATED: 当达到最大池 LUN 数时，报告 free_capacity_gb 为 0。默认情况下，值为 False。
chiscsi_conf = /etc/chelsio-iscsi/chiscsi.conf	字符串值	chiscsi (CXT)全局默认配置文件
cinder_eternus_config_file = /etc/cinder/cinder_fujitsu_eternus_dx.xml	字符串值	cinder eternus_dx 卷驱动程序的配置文件。
cinder_huawei_conf_file = /etc/cinder/cinder_huawei_conf.xml	字符串值	Cinder Huawei 驱动程序的配置文件。
connection_type = iscsi	字符串值	连接到 IBM Storage Array 的连接类型
cycle_period_seconds = 300	整数值	这会定义一个可选周期，它适用于与多循环模式的全局镜像关系。使用 multi_cycling_mode 的一个全局镜像关系最多在每个期间执行一个完整的循环。默认值为 300 秒，有效秒数为 60-86400。
datacore_api_timeout = 300	整数值	从 DataCore API 调用等待响应的秒数。
datacore_disk_failed_delay = 300	整数值	等待 DataCore 虚拟磁盘超过 "Failed" 状态的秒数。
datacore_disk_pools = []	列表值	卷驱动程序可以使用的 DataCore 磁盘镜像列表。
datacore_disk_type = single	字符串值	DataCore 虚拟磁盘类型(single/mirrored)。镜像的虚拟磁盘需要服务器组中的两个存储服务器。

配置选项 = 默认值	类型	Description
datacore_fc_unallowed_targets = []	列表值	无法用于附加卷的 FC 目标列表。要防止 DataCore FibreChannel 卷驱动程序在卷附加中使用一些前端目标，请指定这个选项并列出每个目标的 iqn 和 target 机器作为值，如 <wwpns:target name>, <wwpns:target name>, <wwpns:target name>。
datacore_iscsi_chap_storage = \$state_path/.datacore_chap	字符串值	存储动态生成的 iSCSI CHAP secret 的完全限定文件名。
datacore_iscsi_unallowed_targets = []	列表值	不能用于附加卷的 iSCSI 目标列表。要防止 DataCore iSCSI 卷驱动程序在卷附加中使用一些前端目标，请指定这个选项并列出每个目标的 iqn 和 target 机器作为值，如 <iqn:target name>, <iqn:target name>, <iqn:target name>。
datacore_storage_profile = None	字符串值	DataCore 虚拟磁盘存储配置文件。
default_timeout = 31536000	整数值	CLI 操作的默认超时时间（以分钟为单位）。例如，LUN 迁移是一个典型的长时间运行的操作，它依赖于 LUN 大小和阵列的负载。可以设置特定部署中的上限，以避免不必要的长时间等待。默认情况下，它长度为 365 天。
deferred_deletion_delay = 0	整数值	在标记延迟删除后，卷有资格删除前的时间延迟（以秒为单位）。
deferred_deletion_purge_interval = 60	整数值	运行定期任务之间的秒数，以清除标记要删除的卷。
dell_api_async_rest_timeout = 15	整数值	Dell SC API async 调用默认超时（以秒为单位）。
dell_api_sync_rest_timeout = 30	整数值	Dell SC API 同步调用默认超时（以秒为单位）。
dell_sc_api_port = 3033	端口值	Dell API 端口
dell_sc_server_folder = openstack	字符串值	在 Storage Center 上使用的服务器文件夹的名称
dell_sc_ssn = 64702	整数值	Storage Center 系统串行号
dell_sc_verify_cert = False	布尔值	启用 HTTPS SC SC 证书验证

配置选项 = 默认值	类型	Description
dell_sc_volume_folder = openstack	字符串值	在 Storage Center 上使用的卷文件夹的名称
dell_server_os = Red Hat Linux 6.x	字符串值	在存储中心上创建新服务器时使用的服务器操作系统类型。
destroy_empty_storage_group = False	布尔值	当最后一个 LUN 被删除时，要销毁存储组。默认情况下，值为 False。
disable_discovery = False	布尔值	为 K2 驱动程序上的多路径连接禁用 iSCSI 发现（发送目标）。
<code>`dpl_pool = `</code>	字符串值	存储 DPL 卷的 DPL 池 uuid。
dpl_port = 8357	端口值	DPL 端口号。
driver_client_cert = None	字符串值	如果驱动程序支持客户端证书的路径，进行验证。
driver_client_cert_key = None	字符串值	如果驱动程序支持客户端证书密钥的路径，进行验证。
driver_data_namespace = None	字符串值	要保存到的驱动程序私有数据值的命名空间。
driver_ssl_cert_path = None	字符串值	可以用来指定到带有可信 CA 证书的 CA_BUNDLE 文件或目录的非默认路径，它将用于验证后端
driver_ssl_cert_verify = False	布尔值	如果设置为 True，http 客户端将验证后端端点的 SSL 证书。
driver_use_ssl = False	布尔值	如果驱动程序支持，告诉驱动程序使用 SSL 连接到后端存储。
<code>`ds8k_devadd_unitadd_mapping = `</code>	字符串值	IODevice 地址和单元地址之间的映射。
ds8k_host_type = auto	字符串值	如果您的 OpenStack 版本早于 SRIOV，并且连接到 zLinux 系统，则设置为 zLinux。否则设为 auto。这个参数的有效值为： <i>auto, AMDLinuxRHEL, AMDLinuxSuse, AppleOSX, Fujitsu, Hp, HpTru64, HpVms, LinuxDT, LinuxRF, LinuxRHEL, LinuxSuse, Novell, SGI, SVC, SanFsAIX, SanFsLinux, Sun, VMWare, Win2000, Win2003, Win2008, Win2012, iLinux, nSeries, pLinux, pSeries, pSeriesPowerswap, zLinux, iSeries.</i>
ds8k_ssid_prefix = FF	字符串值	设置 SSID 的前两位数。

配置选项 = 默认值	类型	Description
enable_deferred_deletion = False	布尔值	启用延迟删除。删除后，卷会标记删除，但将在以后异步删除。
enable_unsupported_driver = False	布尔值	当您允许不支持的驱动程序启动时，将其设置为 True。没有维护正常工作的 CI 系统和测试的驱动程序被标记为不受支持，直到 CI 再次工作为止。这也会将驱动程序标记为已弃用，并可在下一发行版本中删除。
enforce_multipath_for_image_xfer = False	布尔值	如果设置为 True，则当 multipathd 未运行时，镜像传输的卷附加将中止。否则，它将回退到单一路径。这个参数需要为每个 backend 部分或 [backend_defaults] 部分中配置，作为所有后端的通用配置。
excluded_domain_ip = None	IP 地址值	DEPRECATED：要在 iSCSI 返回中排除的 Fault Domain IP。弃用自：Stein *reason:* Replaced by excluded_domain_ips 选项
excluded_domain_ips = []	列表值	在 iSCSI 返回中排除的以逗号分隔的 Fault Domain IP。
expiry_thres_minutes = 720	整数值	这个选项指定 NFS 镜像缓存中镜像最后一次访问时间的阈值。当缓存清理循环开始时，在最后 M 分钟内没有访问的镜像（其中 M 是这个参数的值）将从缓存中删除，以便在 NFS 共享上创建可用空间。
extra_capabilities = {}	字符串值	用户定义的功能，即指定键/值对的 JSON 格式的字符串。当请求指定卷类型时，CapabilitiesFilter 可以使用键/值对在后端之间进行选择。例如，指定服务级别或后端的地理位置，然后创建一个卷类型，允许用户由这些不同的属性选择。
filter_function = None	字符串值	将用于过滤主机的模拟字符串。仅在将驱动程序过滤器设置为供 Cinder 调度程序使用时使用。
flashsystem_connection_protocol = FC	字符串值	连接协议应该是 FC。（默认为 FC。）
flashsystem_iscsi_portid = 0	整数值	FlashSystem 的默认 iSCSI 端口 ID。（默认端口为 0。）
flashsystem_multihostmap_enabled = True	布尔值	允许 vdisk 到多主机映射。（默认为 True）

配置选项 = 默认值	类型	Description
force_delete_lun_in_storage_group = True	布尔值	删除 LUN，即使它位于存储组中。
goodness_function = None	字符串值	用于确定主机的好性的模拟字符串。仅在使用良好性 weigher 时使用，供 Cinder 调度程序使用。
gpfs_hosts = []	列表值	以逗号分隔的 GPFS 节点的 IP 地址或主机名列表。
gpfs_hosts_key_file = \$state_path/ssh_known_hosts	字符串值	包含与驱动程序需要与之通信的 gpfs 节点的 SSH 主机密钥的文件。 Default=\$state_path/ssh_known_hosts
gpfs_images_dir = None	字符串值	指定 GPFS 中镜像服务存储库的路径。如果没有将镜像存储在 GPFS 中，则保留未定义。
gpfs_images_share_mode = None	字符串值	指定要使用的镜像副本类型。当镜像服务存储库也使用 GPFS 时设置此选项，以便可以有效地将镜像文件从镜像服务传输到块存储服务。有两个有效值："copy"指定制作镜像的完整副本；"copy_on_write"指定使用 copy-on-write 优化策略，并有效地共享镜像文件的未经修改的块。
gpfs_max_clone_depth = 0	整数值	指定因为快照或克隆而达到特定块所需的间接数量上限。写时复制快照或克隆的冗长链可能会对性能造成负面影响，但提高了空间利用率。0 表示无限克隆深度。
gpfs_mount_point_base = None	字符串值	指定存储块存储卷和快照文件的 GPFS 目录的路径。
<code>`gpfs_private_key = `</code>	字符串值	用于 SSH 验证的私钥的文件名。
gpfs_sparse_volumes = True	布尔值	指定卷是作为稀疏文件创建的，最初消耗了任何空间。如果设置为 False，则该卷将创建为完全分配的文件，在这种情况下，创建可能需要很长时间。
gpfs_ssh_port = 22	端口值	要使用的 SSH 端口。
gpfs_storage_pool = system	字符串值	指定卷要分配给的存储池。默认情况下使用系统存储池。
gpfs_strict_host_key_policy = False	布尔值	在连接到 gpfs 节点时启用严格的 gpfs 主机密钥检查的选项。Default=False
gpfs_user_login = root	字符串值	GPFS 节点的用户名。

配置选项 = 默认值	类型	Description
<code>`gdfs_user_password = `</code>	字符串值	GPFS 节点用户的密码。
<code>hitachi_async_copy_check_interval = 10</code>	整数值	在复制或删除或数据恢复期间检查异步复制状态的时间间隔（以秒为单位）。
<code>hitachi_compute_target_ports = []</code>	列表值	用于将卷附加到计算节点的存储端口 ID。要指定多个端口，请使用逗号连接它们（如 CL1-A、CL2-A）。
<code>hitachi_copy_check_interval = 3</code>	整数值	在卷复制期间检查复制状态的时间间隔（以秒为单位）。
<code>hitachi_copy_speed = 3</code>	整数值	存储系统的副本速度。1 或 2 表示低速度，3 表示中速度，4 到 15 之间的值表示高速度。
<code>hitachi_discard_zero_page = True</code>	布尔值	在 DP-VOL 中启用或禁用零页面回收。
<code>hitachi_exec_retry_interval = 5</code>	整数值	REST API 执行重试间隔（以秒为单位）。
<code>hitachi_extend_timeout = 600</code>	整数值	卷扩展完成的最大等待时间（以秒为单位）。
<code>hitachi_group_create = False</code>	布尔值	如果为 True，则驱动程序将在存储端口上创建主机组或 iSCSI 目标。
<code>hitachi_group_delete = False</code>	布尔值	如果为 True，则驱动程序将根据需要在存储端口上删除主机组或 iSCSI 目标。
<code>hitachi_group_name_format = None</code>	字符串值	主机组、iSCSI 目标和服务器对象的格式。
<code>hitachi_host_mode_options = []</code>	列表值	主机组或 iSCSI 目标的主机模式选项。
<code>hitachi_ldev_range = None</code>	字符串值	LDEV 号的范围，格式为 <code>xxxx-yyyy</code> ，可供驱动程序使用。值可以是十进制格式（如 1000）或以冒号分隔的十六进制格式（如 00:03:E8）。
<code>hitachi_lock_timeout = 7200</code>	整数值	登录或解锁存储的最大等待时间（以秒为单位）。
<code>hitachi_lun_retry_interval = 1</code>	整数值	为 REST API 添加 LUN 映射到服务器的重试间隔（以秒为单位）。

配置选项 = 默认值	类型	Description
hitachi_lun_timeout = 50	整数值	向服务器添加 LUN 映射的最大等待时间（以秒为单位）。
hitachi_pair_target_number = 0	整数值	主机组或 iSCSI 目标的目标名称
hitachi_pools = []	列表值	pool number[s] 或池名称[s] 池的 DP 池。
hitachi_port_scheduler = False	布尔值	向配置的端口启用 WWN 端口，以便 WWN 以轮循方式注册到端口。
hitachi_rest_another_ldev_mapped_retry_timeout = 600	整数值	当新 LUN 分配请求失败时重试时间（以秒为单位）。
hitachi_rest_connect_timeout = 30	整数值	连接到 REST API 会话的最大等待时间（以秒为单位）。
hitachi_rest_disable_io_wait = True	布尔值	此选项将允许立即分离卷。如果设置 False，则存储可能需要几分钟时间在 I/O 后分离卷。
hitachi_rest_get_api_response_timeout = 1800	整数值	针对同步方法的响应的最大等待时间，例如 GET
hitachi_rest_job_api_response_timeout = 1800	整数值	对来自 REST API 的 async 方法的响应的最大等待时间，如 PUT 和 DELETE。
hitachi_rest_keep_session_loop_interval = 180	整数值	保存 REST API 会话的循环间隔（以秒为单位）。
hitachi_rest_pair_target_ports = []	列表值	主机组或 iSCSI 目标对的目标端口名称
hitachi_rest_server_busy_timeout = 7200	整数值	REST API 返回忙碌时的最大等待时间（以秒为单位）。
hitachi_rest_tcp_keepalive = True	布尔值	启用或禁用使用 REST API tcp keepalive
hitachi_rest_tcp_keepcnt = 4	整数值	TCP keepalive 数据包的最大传输数。
hitachi_rest_tcp_keepidle = 60	整数值	等待时间（以秒为单位）发送第一个 TCP keepalive 数据包。
hitachi_rest_tcp_keepintvl = 15	整数值	TCP keepalive 数据包传输间隔（以秒为单位）。

配置选项 = 默认值	类型	Description
hitachi_rest_timeout = 30	整数值	每个 REST API 请求的最大等待时间（以秒为单位）。
hitachi_restore_timeout = 86400	整数值	恢复操作完成的最大等待时间（以秒为单位）。
hitachi_snap_pool = None	字符串值	快照池的池号或池名称。
hitachi_state_transition_timeout = 900	整数值	卷转换完成的最大等待时间（以秒为单位）。
hitachi_storage_id = None	字符串值	存储系统的产品号。
hitachi_target_ports = []	列表值	用于将卷附加到控制器节点的存储端口 ID。要指定多个端口，请使用逗号连接它们（如 CL1-A、CL2-A）。
hitachi_zoning_request = False	布尔值	如果为 True，则驱动程序将在服务器和提供 FC zoning 管理器的存储系统之间配置 FC zoning。
<code>hpe3par_api_url = `</code>	字符串值	WSAPI 服务器 URL。此设置适用于 3PAR 和 Primera。 示例 1: 对于 3PAR, URL 是: <a href="https://<3par ip>:8080/api/v1">https://<3par ip>:8080/api/v1 Example 2: for Primera, URL is: <a href="https://<primera ip>:443/api/v1">https://<primera ip>:443/api/v1
hpe3par_cpg = ['OpenStack']	列表值	用于卷创建的 3PAR / Primera CPG 列表
<code>hpe3par_cpg_snap = `</code>	字符串值	用于卷快照的 3PAR / Primera CPG。如果为空，则将使用 userCPG。
hpe3par_debug = False	布尔值	启用到 3PAR / Primera 的 HTTP 调试
hpe3par_iscsi_chap_enabled = False	布尔值	为 iSCSI 连接启用 CHAP 验证。
hpe3par_iscsi_ips = []	列表值	要使用的目标 iSCSI 地址列表。
<code>hpe3par_password = `</code>	字符串值	在 hpe3par_username 中指定的用户的 3PAR / Primera 密码
<code>hpe3par_snapshot_expiration = `</code>	字符串值	快照过期并被删除的时间（以小时为单位）。这必须大于过期

配置选项 = 默认值	类型	Description
<code>hpe3par_snapshot_retention = `</code>	字符串值	保留快照的时间（以小时为单位）。在过期前您无法删除它。
<code>hpe3par_target_nsp = `</code>	字符串值	当 (1) cinder.conf 中未启用多路径时，要使用的 3PAR 后端的 nsp。(2)光纤通道区域管理器未使用。3PAR 后端只适用于这个特定的 nsp。例如，如果 nsp 是 212，则该选项的值的格式是 2:1:2
<code>hpe3par_username = `</code>	字符串值	3PAR / Primera 用户名及 <i>edit</i> 角色
<code>hpexp_async_copy_check_interval = 10</code>	整数值	异步检查复制的时间间隔（以秒为单位）
<code>hpexp_compute_target_ports = []</code>	列表值	用于将卷附加到计算节点的存储端口 ID。要指定多个端口，请使用逗号连接它们（如 CL1-A、CL2-A）。
<code>hpexp_copy_check_interval = 3</code>	整数值	检查复制的时间间隔（以秒为单位）
<code>hpexp_copy_speed = 3</code>	整数值	存储系统的副本速度。1 或 2 表示低速度，3 表示中速度，4 到 15 之间的值表示高速度。
<code>hpexp_discard_zero_page = True</code>	布尔值	在 THP V-VOL 中启用或禁用零页面回收。
<code>hpexp_exec_retry_interval = 5</code>	整数值	REST API 执行重试间隔（以秒为单位）。
<code>hpexp_extend_timeout = 600</code>	整数值	卷扩展完成的最大等待时间（以秒为单位）。
<code>hpexp_group_create = False</code>	布尔值	如果为 True，则驱动程序将在存储端口上创建主机组或 iSCSI 目标。
<code>hpexp_group_delete = False</code>	布尔值	如果为 True，则驱动程序将根据需要在存储端口上删除主机组或 iSCSI 目标。
<code>hpexp_host_mode_options = []</code>	列表值	主机组或 iSCSI 目标的主机模式选项。
<code>hpexp_ldev_range = None</code>	字符串值	LDEV 号的范围，格式为 xxxx-yyyy，可供驱动程序使用。值可以是十进制格式（如 1000）或以冒号分隔的十六进制格式（如 00:03:E8）。
<code>hpexp_lock_timeout = 7200</code>	整数值	解锁存储的最大等待时间（以秒为单位）。

配置选项 = 默认值	类型	Description
hpexp_lun_retry_interval = 1	整数值	添加 LUN 的 REST API 重试间隔（以秒为单位）。
hpexp_lun_timeout = 50	整数值	添加 LUN 完成的最大等待时间（以秒为单位）。
hpexp_pools = []	列表值	THP 池的池名称[s] 或池名称。
hpexp_rest_another_ldev_mapped_retry_timeout = 600	整数值	当新 LUN 分配请求失败时重试时间（以秒为单位）。
hpexp_rest_connect_timeout = 30	整数值	REST API 连接完成的最大等待时间（以秒为单位）。
hpexp_rest_disable_io_wait = True	布尔值	在 I/O 后，可能需要过些时间来分离卷。此选项将允许断开卷立即完成。
hpexp_rest_get_api_response_timeout = 1800	整数值	对 REST API 的 GET 方法的响应的最大等待时间（以秒为单位）。
hpexp_rest_job_api_response_timeout = 1800	整数值	从 REST API 响应的最大等待时间（以秒为单位）。
hpexp_rest_keep_session_loop_interval = 180	整数值	保存 REST API 会话的循环间隔（以秒为单位）。
hpexp_rest_server_busy_timeout = 7200	整数值	REST API 返回忙碌时的最大等待时间（以秒为单位）。
hpexp_rest_tcp_keepalive = True	布尔值	启用或禁用使用 REST API tcp keepalive
hpexp_rest_tcp_keepcnt = 4	整数值	TCP keepalive 数据包的最大传输数。
hpexp_rest_tcp_keepidle = 60	整数值	等待时间（以秒为单位）发送第一个 TCP keepalive 数据包。
hpexp_rest_tcp_keepintvl = 15	整数值	TCP keepalive 数据包传输间隔（以秒为单位）。
hpexp_rest_timeout = 30	整数值	REST API 执行完成的最大等待时间（以秒为单位）。
hpexp_restore_timeout = 86400	整数值	恢复操作完成的最大等待时间（以秒为单位）。
hpexp_snap_pool = None	字符串值	快照池的池号或池名称。

配置选项 = 默认值	类型	Description
hpexp_state_transition_timeout = 900	整数值	卷转换完成的最大等待时间（以秒为单位）。
hpexp_storage_id = None	字符串值	存储系统的产品号。
hpexp_target_ports = []	列表值	用于将卷附加到控制器节点的存储端口 ID。要指定多个端口，请使用逗号连接它们（如 CL1-A、CL2-A）。
hpexp_zoning_request = False	布尔值	如果为 True，则驱动程序将在服务器和提供 FC zoning 管理器的存储系统之间配置 FC zoning。
hpmsa_api_protocol = https	字符串值	HPMSA API 接口协议。
hpmsa_iscsi_ips = []	列表值	以逗号分隔的目标 iSCSI IP 地址列表。
hpmsa_pool_name = A	字符串值	用于创建卷的池或 Vdisk 名称。
hpmsa_pool_type = virtual	字符串值	线性（用于 Vdisk）或虚拟（池）。
hpmsa_verify_certificate = False	布尔值	是否验证 HPMSA 阵列 SSL 证书。
hpmsa_verify_certificate_path = None	字符串值	HPMSA 阵列 SSL 证书路径。
hypermetro_devices = None	字符串值	远程设备 hypermetro 将使用。
iet_conf = /etc/iet/ietd.conf	字符串值	DEPRECATED: IET 配置文件
ignore_pool_full_threshold = False	布尔值	强制创建 LUN，即使达到池的完整阈值。默认情况下，值为 False。
image_upload_use_cinder_backend = False	布尔值	如果设置为 True，则以原始格式的 upload-to-image 将创建一个克隆的卷，并将其位置注册到镜像服务，而不是上传卷内容。在镜像服务中必须启用 cinder 后端和位置支持。
image_upload_use_internal_tenant = False	布尔值	如果设置为 True，则由 upload-to-image 创建的镜像卷将放入内部租户中。否则，镜像卷会在当前上下文的租户中创建。

配置选项 = 默认值	类型	Description
image_volume_cache_enabled = False	布尔值	为此后端启用镜像卷缓存。
image_volume_cache_max_count = 0	整数值	镜像卷缓存中允许的最大条目数。0 jpeg unlimited。
image_volume_cache_max_size_gb = 0	整数值	此后端的镜像卷缓存的最大大小（以 GB 为单位）。0 jpeg unlimited。
included_domain_ips = []	列表值	从 iSCSI 返回中以逗号分隔的 Fault 域 IP。
infinidat_iscsi_netspaces = []	列表值	用于 iSCSI 连接的网络空格名称列表
infinidat_pool_name = None	字符串值	从中分配卷的池名称
infinidat_storage_protocol = fc	字符串值	在主机和存储后端之间传输数据的协议。
infinidat_use_compression = False	布尔值	指定是否为新创建的卷打开压缩。
initiator_auto_deregistration = False	布尔值	在相关存储组销毁后自动取消注册启动器。默认情况下，值为 False。
initiator_auto_registration = False	布尔值	自动注册发起程序。默认情况下，值为 False。
initiator_check = False	布尔值	使用这个值启用 initiator_check。
interval = 3	整数值	使用这个值指定间隔长度（以秒为单位）。
io_port_list = None	列表值	在 Nova 或 Cinder 中使用逗号分隔的 iSCSI 或 FC 端口。
iscsi_initiators = None	字符串值	在主机名及其 iSCSI 启动器 IP 地址之间进行映射。
iscsi_iotype = fileio	字符串值	将 iSCSI 目标的行为设置为执行 blockio 或 fileio，可以设置 auto，Cinder 将自动检测后备设备的类型
`iscsi_target_flags = `	字符串值	为 iSCSI 目标设置特定于目标的标记。仅用于 tgtadm 来使用 bsoflags 选项指定后备设备标志。指定的字符串被传递为底层工具。

配置选项 = 默认值	类型	Description
iscsi_write_cache = on	字符串值	将 iSCSI 目标的行为设置为执行回写(on)或 write-through (off)。如果 target_helper 设置为 tgtadm, 则此参数有效。
iser_helper = tgtadm	字符串值	要使用的 iSER 目标用户固定工具的名称
iser_ip_address = \$my_ip	字符串值	iSER 守护进程侦听的 IP 地址
iser_port = 3260	端口值	iSER 守护进程侦听的端口
iser_target_prefix = iqn.2010-10.org.openstack:	字符串值	iSER 卷的前缀
lenovo_api_protocol = https	字符串值	Lenovo api 接口协议。
lenovo_iscsi_ips = []	列表值	以逗号分隔的目标 iSCSI IP 地址列表。
lenovo_pool_name = A	字符串值	用于创建卷的池或 Vdisk 名称。
lenovo_pool_type = virtual	字符串值	线性（用于 VDisk）或虚拟（池）。
lenovo_verify_certificate = False	布尔值	是否验证 Lenovo 阵列 SSL 证书。
lenovo_verify_certificate_path = None	字符串值	Lenovo 阵列 SSL 证书路径。
linstor_autoplace_count = 0	整数值	卷部署的自动扩展复制计数。0 = 完整集群复制而没有自动地进行, 1 = 单节点部署, 没有复制, 2 或更高 = 使用自动原位部署复制。
linstor_controller_diskless = True	布尔值	True 表示 Cinder 节点是一个无盘 LINSTOR 节点。
linstor_default_blocksize = 4096	整数值	镜像恢复的默认块大小。使用 iSCSI 传输时, 这个选项指定块大小。
linstor_default_storage_pool_name = DfltStorPool	字符串值	LINSTOR 的默认存储池名称。
linstor_default_uri = linstor://localhost	字符串值	LINSTOR 的默认存储 URI。

配置选项 = 默认值	类型	Description
linstor_default_volume_group_name = drbd-vg	字符串值	LINSTOR 的默认卷组名称。不是 Cinder 卷。
linstor_volume_downsize_factor = 4096	浮点值	以 KiB 为单位的默认卷缩减的大小 = 4 MiB。
load_balance = False	布尔值	为 PowerMax 后端启用/禁用负载均衡。
load_balance_real_time = False	布尔值	为 PowerMax 后端的端口级别负载均衡启用/禁用实时性能指标。
load_data_format = Avg	字符串值	性能数据格式，不适用于实时指标。可用选项包括 "avg" 和 "max"。
load_look_back = 60	整数值	在负载计算中查询诊断性能指标的时间，最少为 0 个 1440 (24 小时)。
load_look_back_real_time = 1	整数值	在负载计算中查询实时性能指标的时间（以分钟为单位），最少为 10 个。
lss_range_for_cg = `	字符串值	为一致性组保留 LSS。
lvm_conf_file = /etc/cinder/lvm.conf	字符串值	在 Cinder 中用于 LVM 驱动程序的 LVM conf 文件；如果指定的文件不存在，则忽略此设置（您也可以指定 <i>None</i> 来不使用 conf 文件（即使存在的话）。
lvm_mirrors = 0	整数值	如果 >0，则使用多个镜像创建 LV。请注意，这需要 lvm_mirrors + 2 个带有可用空间的 PV
lvm_share_target = False	布尔值	是否为所有 LUN 共享相同的目标（当前只支持 nvmet。
lvm_suppress_fd_warnings = False	布尔值	抑制 LVM 命令中泄漏的文件描述符警告。
lvm_type = auto	字符串值	要部署的 LVM 卷类型；（默认、精简或 auto）。如果支持精简，则 auto 默认为 thin。
macrosan_client = None	列表值	Macrosan iscsi_clients 列表。您可以配置多个客户端。您可以使用以下格式进行配置：(host; client_name; sp1_iscsi_port; sp2_iscsi_port),(host; client_name; sp1_iscsi_port; sp2_iscsi_port) Important 警告，client_name 具有以下要求：[a-zA-Z0-9.-_:]，最大字符数为 31 E.g: (controller1; device1; eth-1:0; eth-2:0),(controller2; device2; eth-1:0/eth-1:1; eth-2:0/eth-2:1),

配置选项 = 默认值	类型	Description
macrosan_client_default = None	字符串值	这是 iscsi 的默认连接端口的名称。当没有获取主机相关的信息时，会使用此默认配置。例如：eth-1:0/eth-1:1; eth-2:0/eth-2:1
macrosan_fc_keep_mapped_ports = True	布尔值	如果是 FC 连接，则会维护与端口关联的配置项目。
macrosan_fc_use_sp_port_nr = 1	整数值	use_sp_port_nr 参数是在交换机非所有传递模式下建立 FC 连接时，单行内存使用的在线 FC 端口数量。最大值为 4
macrosan_force_unmap_initl = True	布尔值	在删除卷时强制断开连接
macrosan_log_timing = True	布尔值	是否启用日志时间
macrosan_pool = None	字符串值	用于卷创建的池
macrosan_replication_destination_ports = None	列表值	从设备
macrosan_replication_ipaddrs = None	列表值	MacroSAN 复制设备的 ip 地址
macrosan_replication_password = None	字符串值	MacroSAN 复制设备的密码
macrosan_replication_username = None	字符串值	MacroSAN 复制设备的用户名
macrosan_sdas_ipaddrs = None	列表值	MacroSAN sdas 设备的 IP 地址
macrosan_sdas_password = None	字符串值	MacroSAN sdas 设备的密码
macrosan_sdas_username = None	字符串值	MacroSAN sdas 设备的用户名
macrosan_snapshot_resource_ratio = 1.0	浮点值	设置快照的资源比率
macrosan_thin_lun_extents_size = 8	整数值	设置 thin lun 的扩展大小

配置选项 = 默认值	类型	Description
macrosan_thin_lun_high_watermark = 20	整数值	设置 thin lun 的高水位线
macrosan_thin_lun_low_watermark = 5	整数值	设置 thin lun 的低水位线
<code>`management_ips = `</code>	字符串值	管理 IP 地址列表（用逗号分开）
max_luns_per_storage_group = 255	整数值	存储组中 LUN 的默认最大数目。默认情况下，值为 255。
max_over_subscription_ratio = 20.0	字符串值	启用精简配置时，超订阅比率的表示。默认比率为 20.0，这意味着置备的容量可以是总物理容量的 20 倍。如果比率是 10.5，这表示调配的容量可以是总物理容量的 10.5 倍。比率 1.0 表示调配的容量不能超过物理容量的总容量。如果比率是 <i>auto</i> ，Cinder 将根据调配的容量和已用空间自动计算比率。如果没有设置 <i>auto</i> ，则比率必须至少为 1.0。
metro_domain_name = None	字符串值	远程 metro 设备域名。
metro_san_address = None	字符串值	远程 metro 设备请求 url。
metro_san_password = None	字符串值	远程 metro 设备 san 密码。
metro_san_user = None	字符串值	远程 metro 设备 san 用户。
metro_storage_pools = None	字符串值	远程 metro 设备池名称。
<code>`nas_host = `</code>	字符串值	NAS 系统的 IP 地址或主机名。
nas_login = admin	字符串值	连接到 NAS 系统的用户名。
nas_mount_options = None	字符串值	用于挂载 Cinder 卷的存储后端文件系统的选项。
<code>`nas_password = `</code>	字符串值	连接到 NAS 系统的密码。
<code>`nas_private_key = `</code>	字符串值	用于 SSH 验证的私钥的文件名。

配置选项 = 默认值	类型	Description
nas_secure_file_operations = auto	字符串值	允许网络附加存储系统在不允许 root 级别访问的安全环境中操作。如果设置为 False，则访问权限是 root 用户和 insecure。如果设置为 True，则访问不是 root。如果设置为 auto，则进行检查来确定这是新的安装：如果是新的安装，则使用 True，否则为 False。默认为 auto。
nas_secure_file_permissions = auto	字符串值	在网络附加存储卷文件中设置更安全的文件权限，以限制广泛的其他/全局访问。如果设置为 False，则使用打开权限创建卷。如果设置为 True，则使用 cinder 用户和组(660)的权限创建卷。如果设置为 auto，则进行检查来确定这是新的安装：如果是新的安装，则使用 True，否则为 False。默认为 auto。
<code>nas_share_path = `</code>	字符串值	用于存储 Cinder 卷的共享的路径。例如："/srv/export1"用于位于 10.0.5.10:/srv/export1 的 NFS 服务器导出。
nas_ssh_port = 22	端口值	用于连接到 NAS 系统的 SSH 端口。
nas_volume_prov_type = thin	字符串值	创建卷时使用的调配类型。
naviseccli_path = None	字符串值	Naviseccli 路径。
nec_v_async_copy_check_interval = 10	整数值	在复制或删除或数据恢复期间检查异步复制状态的时间间隔（以秒为单位）。
nec_v_compute_target_ports = []	列表值	用于将卷附加到计算节点的存储端口 ID。要指定多个端口，请使用逗号连接它们（如 CL1-A、CL2-A）。
nec_v_copy_check_interval = 3	整数值	在卷复制期间检查复制状态的时间间隔（以秒为单位）。
nec_v_copy_speed = 3	整数值	存储系统的副本速度。1 或 2 表示低速度，3 表示中速度，4 到 15 之间的值表示高速度。
nec_v_discard_zero_page = True	布尔值	在 DP-VOL 中启用或禁用零页面回收。
nec_v_exec_retry_interval = 5	整数值	REST API 执行重试间隔（以秒为单位）。
nec_v_extend_timeout = 600	整数值	卷扩展完成的最大等待时间（以秒为单位）。
nec_v_group_create = False	布尔值	如果为 True，则驱动程序将在存储端口上创建主机组或 iSCSI 目标。

配置选项 = 默认值	类型	Description
nec_v_group_delete = False	布尔值	如果为 True，则驱动程序将根据需要在存储端口上删除主机组或 iSCSI 目标。
nec_v_host_mode_options = []	列表值	主机组或 iSCSI 目标的主机模式选项
nec_v_ldev_range = None	字符串值	LDEV 号的范围，格式为 xxxx-yyyy，可供驱动程序使用。值可以是十进制格式（如 1000）或以冒号分隔的十六进制格式（如 00:03:E8）。
nec_v_lock_timeout = 7200	整数值	解锁存储的最大等待时间（以秒为单位）。
nec_v_lun_retry_interval = 1	整数值	添加 LUN 的 REST API 重试间隔（以秒为单位）。
nec_v_lun_timeout = 50	整数值	添加 LUN 完成的最大等待时间（以秒为单位）。
nec_v_pools = []	列表值	pool number[s] 或池名称[s] 池的 DP 池。
nec_v_rest_another_ldev_mapped_retry_timeout = 600	整数值	当新 LUN 分配请求失败时重试时间（以秒为单位）。
nec_v_rest_connect_timeout = 30	整数值	REST API 连接完成的最大等待时间（以秒为单位）。
nec_v_rest_disable_io_wait = True	布尔值	在 I/O 后，可能需要过些时间来分离卷。此选项将允许断开卷立即完成。
nec_v_rest_get_api_response_timeout = 1800	整数值	对 REST API 的 GET 方法的响应的最大等待时间（以秒为单位）。
nec_v_rest_job_api_response_timeout = 1800	整数值	从 REST API 响应的最大等待时间（以秒为单位）。
nec_v_rest_keep_session_loop_interval = 180	整数值	保存 REST API 会话的循环间隔（以秒为单位）。
nec_v_rest_server_busy_timeout = 7200	整数值	REST API 返回忙碌时的最大等待时间（以秒为单位）。
nec_v_rest_tcp_keepalive = True	布尔值	启用或禁用使用 REST API tcp keepalive
nec_v_rest_tcp_keepcnt = 4	整数值	TCP keepalive 数据包的最大传输数。

配置选项 = 默认值	类型	Description
nec_v_rest_tcp_keepidle = 60	整数值	等待时间（以秒为单位）发送第一个 TCP keepalive 数据包。
nec_v_rest_tcp_keepintvl = 15	整数值	TCP keepalive 数据包传输间隔（以秒为单位）。
nec_v_rest_timeout = 30	整数值	REST API 执行完成的最大等待时间（以秒为单位）。
nec_v_restore_timeout = 86400	整数值	恢复操作完成的最大等待时间（以秒为单位）。
nec_v_snap_pool = None	字符串值	快照池的池号或池名称。
nec_v_state_transition_timeout = 900	整数值	卷转换完成的最大等待时间（以秒为单位）。
nec_v_storage_id = None	字符串值	存储系统的产品号。
nec_v_target_ports = []	列表值	用于将卷附加到控制器节点的存储端口 ID。要指定多个端口，请使用逗号连接它们（如 CL1-A、CL2-A）。
nec_v_zoning_request = False	布尔值	如果为 True，则驱动程序将在服务器和提供 FC zoning 管理器的存储系统之间配置 FC zoning。
netapp_api_trace_pattern = (.*)	字符串值	限制 API 跟踪的正则表达式。只有在使用 trace_flags 选项启用 api 追踪时，才会接受这个选项。默认情况下，所有 API 都会被 traced。
netapp_copyoffload_tool_path = None	字符串值	这个选项指定 NetApp copy offload 工具二进制的路径。确保二进制文件设置了执行权限，以允许 cinder-volume 进程的有效用户执行该文件。
netapp_host_type = None	字符串值	这个选项定义了可以访问 LUN 的所有启动器的操作系统类型。此信息在将 LUN 映射到单个主机或主机组时使用。
netapp_login = None	字符串值	用于访问存储系统或代理服务器的管理用户帐户名称。
netapp_lun_ostype = None	字符串值	此选项定义将访问从 Data ONTAP 导出的 LUN 的操作系统类型；在创建时分配给 LUN。
netapp_lun_space_reservation = enabled	字符串值	这个选项确定是否为 LUN 分配保留存储空间。如果启用，LUN 会被置备。如果禁用了空间保留，则按需分配存储空间。

配置选项 = 默认值	类型	Description
netapp_nfs_image_cache_cleanup_interval = 600	整数值	设置 NFS 镜像缓存清理任务之间的时间（以秒为单位）。
netapp_password = None	字符串值	netapp_login 选项中指定的管理用户帐户的密码。
netapp_pool_name_search_pattern = (.+)	字符串值	此选项用于将置备限制到指定的池。将此选项的值指定为正则表达式，它将应用到代表 Cinder 中池的存储后端的对象名称。只有在存储协议被配置为使用 iSCSI 或 FC 时，才会使用这个选项。
netapp_replication_aggregate_map = None	dict 值	使用整个后端复制时，多选择字典来代表源和目标后端之间的聚合映射。对于与 cinder 池关联的每个源聚合（NetApp FlexVol/FlexGroup，您需要在复制目标设备上指定目标聚合。复制目标设备使用配置选项 replication_device 配置。将这个选项指定为您复制设备的次数。每个条目都使用标准字典配置表单： netapp_replication_aggregate_map = backend_id: <name_of_replication_device_section>,src_aggr_name1:dest_aggr_name1,src_aggr_name2:dest_aggr_name2,...
netapp_replication_volume_online_timeout = 360	整数值	设置等待复制卷创建完成并在线的时间（以秒为单位）。
netapp_server_hostname = None	字符串值	存储系统或代理服务器的主机名（或 IP 地址）。
netapp_server_port = None	整数值	用于与存储系统或代理服务器通信的 TCP 端口。如果没有指定，Data ONTAP 驱动程序将使用 80 用于 HTTP，443 用于 HTTPS。
netapp_size_multiplier = 1.2	浮点值	请求的大小乘以请求的卷大小，以确保虚拟存储服务器(Vserver)上有足够的空间来满足卷创建请求。注：这个选项已弃用，并将在 Mitaka 发行版本中使用 "reserved_percentage" 删除。
netapp_snapmirror_quiesce_timeout = 3600	整数值	在故障转移期间中止前，等待现有 SnapMirror 传输完成的最长时间（以秒为单位）。
netapp_storage_family = ontap_cluster	字符串值	存储系统上使用的 storage 系列类型；唯一有效的值是 ontap_cluster，以便使用集群数据 ONTAP。
netapp_storage_protocol = None	字符串值	在存储系统的数据路径中使用的存储协议。
netapp_transport_type = http	字符串值	与存储系统或代理服务器通信时使用的传输协议。

配置选项 = 默认值	类型	Description
netapp_vserver = None	字符串值	这个选项在存储集群中指定虚拟存储服务器(Vserver)名称, 其中应置备块存储卷。
nexenta_blocksize = 4096	整数值	数据集的块大小
nexenta_chunksize = 32768	整数值	NexentaEdge iSCSI LUN 对象块大小
<code>`nexenta_client_address = `</code>	字符串值	非 VIP 服务的 NexentaEdge iSCSI 网关客户端地址
nexenta_dataset_compression = on	字符串值	新 ZFS 文件夹的压缩值。
nexenta_dataset_dedup = off	字符串值	新 ZFS 文件夹的 deduplication 值。
<code>`nexenta_dataset_description = `</code>	字符串值	文件夹人类可读的描述。
nexenta_encryption = False	布尔值	定义是否启用了 NexentaEdge iSCSI LUN 对象。
<code>`nexenta_folder = `</code>	字符串值	cinder 创建的数据集所在的文件夹。
nexenta_group_snapshot_template = group-snapshot-%s	字符串值	生成组群快照名称的模板字符串
<code>`nexenta_host = `</code>	字符串值	NexentaStor 设备的 IP 地址
nexenta_host_group_prefix = cinder	字符串值	NexentaStor 上 iSCSI 主机组的前缀
nexenta_iops_limit = 0	整数值	NexentaEdge iSCSI LUN 对象 IOPS 限制
<code>`nexenta_iscsi_service = `</code>	字符串值	NexentaEdge iSCSI 服务名称
nexenta_iscsi_target_host_group = all	字符串值	允许访问卷的主机组
<code>`nexenta_iscsi_target_portal_groups = `</code>	字符串值	NexentaStor 目标门户组
nexenta_iscsi_target_portal_port = 3260	整数值	nexenta 设备 iSCSI 目标门户端口

配置选项 = 默认值	类型	Description
<code>`nexenta_iscsi_target_portals = `</code>	字符串值	NexentaStor5 的以逗号分隔的门户列表，格式为 IP1:port1,IP2:port2。port 是可选的，default=3260。 实例：10.10.10.1:3267,10.10.1.2
<code>nexenta_lu_writebackcache_disabled = False</code>	布尔值	后备存储或非后写入
<code>`nexenta_lun_container = `</code>	字符串值	LUN 的 NexentaEdge 逻辑路径
<code>nexenta_luns_per_target = 100</code>	整数值	每个 iSCSI 目标的 LUN 数量
<code>nexenta_mount_point_base = \$state_path/mnt</code>	字符串值	包含 NFS 共享挂载点的基础目录
<code>nexenta_nbd_symlinks_dir = /dev/disk/by-path</code>	字符串值	NexentaEdge 逻辑路径，用于存储到 NBD 的符号链接
<code>nexenta_nms_cache_volroot = True</code>	布尔值	如果设置 True cache NexentaStor 设备 volroot 选项值。
<code>nexenta_ns5_blocksize = 32</code>	整数值	数据集的块大小
<code>nexenta_origin_snapshot_template = origin-snapshot-%s</code>	字符串值	生成克隆的源名称的模板字符串
<code>nexenta_password = nexenta</code>	字符串值	连接到 NexentaStor 管理 REST API 服务器的密码
<code>nexenta_qcow2_volumes = False</code>	布尔值	将卷创建为 QCOW2 文件而不是 raw 文件
<code>nexenta_replication_count = 3</code>	整数值	NexentaEdge iSCSI LUN 对象复制计数。
<code>`nexenta_rest_address = `</code>	字符串值	NexentaStor 管理 REST API 端点的 IP 地址
<code>nexenta_rest_backoff_factor = 0.5</code>	浮点值	指定在连接尝试 NexentaStor 管理 REST API 服务器之间应用的 backoff 因素
<code>nexenta_rest_connect_timeout = 30</code>	浮点值	指定必须与 NexentaStor 管理 REST API 服务器的连接的时间限制（以秒为单位）
<code>nexenta_rest_password = nexenta</code>	字符串值	连接到 NexentaEdge 的密码。

配置选项 = 默认值	类型	Description
nexenta_rest_port = 0	整数值	用于连接到 NexentaStor 管理 REST API 服务器的 HTTP(S) 端口。如果它相等为零，则使用 HTTPS 为 8443，使用 8080 用于 HTTP
nexenta_rest_protocol = auto	字符串值	使用 http 或 https 进行 NexentaStor 管理 REST API 连接（默认 auto）
nexenta_rest_read_timeout = 300	浮点值	指定 NexentaStor 管理 REST API 服务器必须发送响应的时间限制（以秒为单位）
nexenta_rest_retry_count = 3	整数值	指定在连接错误和 NexentaStor 设备 EBUSY 或 ENOENT 错误时重复 NexentaStor 管理 REST API 调用的次数
nexenta_rest_user = admin	字符串值	连接到 NexentaEdge 的用户名。
nexenta_rrmgr_compression = 0	整数值	启用流压缩，级别 1.9.1 - 提供最佳速度；9 - 提供最佳压缩。
nexenta_rrmgr_connections = 2	整数值	TCP 连接数。
nexenta_rrmgr_tcp_buffer_size = 4096	整数值	KiloBytes 中的 TCP 缓冲区大小。
nexenta_shares_config = /etc/cinder/nfs_shares	字符串值	带有可用 nfs 共享列表的文件
nexenta_sparse = False	布尔值	启用或禁用稀疏数据集的创建
nexenta_sparsed_volumes = True	布尔值	启用或禁用将卷创建为没有空间的稀疏文件。如果禁用(False)，则卷作为常规文件创建，这需要很长时间。
nexenta_target_group_prefix = cinder	字符串值	NexentaStor 上 iSCSI 目标组的前缀
nexenta_target_prefix = iqn.1986-03.com.sun:02:cinder	字符串值	NexentaStor iSCSI 目标的 IQN 前缀
nexenta_use_https = True	布尔值	使用 HTTP 安全协议进行 NexentaStor 管理 REST API 连接
nexenta_user = admin	字符串值	连接到 NexentaStor 管理 REST API 服务器的用户名

配置选项 = 默认值	类型	Description
nexenta_volume = cinder	字符串值	包含所有卷的 NexentaStor 池名称
nexenta_volume_group = iscsi	字符串值	NexentaStor5 iSCSI 的卷组
nfs_mount_attempts = 3	整数值	在引发错误前挂载 NFS 共享的次数。最少要尝试挂载 NFS 共享，无论指定的值如何。
nfs_mount_options = None	字符串值	传递给 NFS 客户端的挂载选项。详情请查看 NFS (5) 手册页。
nfs_mount_point_base = \$state_path/mnt	字符串值	包含 NFS 共享的挂载点的基本目录。
nfs_qcow2_volumes = False	布尔值	将卷创建为 QCOW2 文件，而不是 raw 文件。
nfs_shares_config = /etc/cinder/nfs_shares	字符串值	包含可用 NFS 共享列表的文件。
nfs_snapshot_support = False	布尔值	启用对 NFS 驱动程序上的快照的支持。使用 libvirt <1.2.7 的平台将遇到此功能的问题。
nfs_sparsed_volumes = True	布尔值	将卷创建为稀疏文件，使其没有空间。如果设置为 False 卷，则作为常规文件创建。在这种情况下，卷创建需要大量时间。
nimble_pool_name = default	字符串值	nimble Controller 池名称
nimble_subnet_label = *	字符串值	nimble 子网标签
nimble_verify_cert_path = None	字符串值	Nimble Array SSL 证书的路径
nimble_verify_certificate = False	布尔值	是否验证 Nimble SSL 证书
num_iser_scan_tries = 3	整数值	重新扫描 iSER 目标查找卷的次数上限
num_shell_tries = 3	整数值	尝试运行 flakey shell 命令的次数
num_volume_device_scan_tries = 3	整数值	重新扫描目标以查找卷的次数上限

配置选项 = 默认值	类型	Description
nvmeof_conn_info_version = 1	整数值	NVMe os-brick 连接器有 2 种不同的连接信息格式，这允许一些使用原始格式（版本 1）的 NVMe-oF 驱动程序（如 spdk 和 LVM-nvmet）来发送更新的格式。
nvmet_ns_id = 10	整数值	没有共享目标时 LVM 卷的子系统的命名空间 ID。共享时的最小 id 值。Linux 中支持的最大值为 8192
nvmet_port_id = 1	端口值	没有共享目标时 NVMe 目标端口定义的 id。共享时起始端口 id 值，为每个二级 ip 地址递增。
port_group_load_metric = PercentBusy	字符串值	用于端口组负载计算的指标。
port_load_metric = PercentBusy	字符串值	用于端口负载计算的指标。
powerflex_allow_migration_during_rebuild = False	布尔值	允许重建过程中进行卷迁移。
powerflex_allow_non_padded_volumes = False	布尔值	当禁用零填充时，允许在存储池中创建卷。如果多个租户将使用来自共享存储池中的卷，则不应启用此选项。
powerflex_max_over_subscription_ratio = 10.0	浮点值	驱动程序的 max_over_subscription_ratio 设置。允许的最大值为 10.0。
powerflex_rest_server_port = 443	端口值	Gateway REST 服务器端口。
powerflex_round_volume_capacity = True	布尔值	将卷大小向上舍入到 8GB 边界。powerflex/VxFlex OS 需要以 8GB 的倍数进行卷大小。如果设置为 False，则卷创建将无法正确大小
powerflex_server_api_version = None	字符串值	powerflex/ScaleIO API 版本。除非由技术支持指示，否则该值应保留为默认值。
powerflex_storage_pools = None	字符串值	存储池.用于提供卷的、以逗号分隔的存储池列表。每个池应指定为 protection_domain_name:storage_pool_name 值
powerflex_unmap_volume_before_deletion = False	布尔值	在删除前取消映射卷。
powermax_array = None	字符串值	要连接的阵列的序列号。
powermax_array_tag_list = None	列表值	存储阵列的用户分配名称列表。

配置选项 = 默认值	类型	Description
powermax_port_group_name_template = portGroupName	字符串值	为端口组名称定义的覆盖。
powermax_port_groups = None	列表值	包含在服务器连接之前配置前端端口的端口组列表。
powermax_service_level = None	字符串值	用于置备存储的服务级别。把它设置为 pool_name 中的额外规格。
powermax_short_host_name_template = shortHostName	字符串值	用户定义的用于短主机名的覆盖。
powermax_srp = None	字符串值	用于置备的阵列上的存储资源池。
powerstore_appliances = []	列表值	设备名称。用于置备卷的 PowerStore 设备名称的逗号分隔列表。弃用自：Wallaby *reason:*Is 不再被使用。PowerStore Load Balancer 用于置备卷。
powerstore_ports = []	列表值	允许的端口。以逗号分隔的 PowerStore iSCSI IP 或 FC WWN（例如要使用的 58:CC:f0:98:49:22:07:02）。如果没有设置选项，则允许所有端口。
proxy = cinder.volume.drivers.ibm.ibm_storage.proxy.IBMStorageProxy	字符串值	连接到 IBM Storage Array 的代理驱动程序
pure_api_token = None	字符串值	REST API 身份验证令牌。
pure_automatic_max_over_subscription_ratio = True	布尔值	根据当前的数据缩减值自动决定超额订阅比率。如果使用这个计算的值，将覆盖 max_over_subscription_ratio 配置选项。
pure_eradicate_on_delete = False	布尔值	启用后，所有 Pure 卷、快照和保护组将在 Cinder 中删除时分配。删除此设置为 True 后，数据将无法恢复！禁用后，卷和快照将进入待处理的 eradication 状态，并可恢复。
pure_host_personality = None	字符串值	决定 Purity 系统如何调整阵列和启动器之间使用的协议。

配置选项 = 默认值	类型	Description
pure_iscsi_cidr = 0.0.0.0/0	字符串值	FlashArray iSCSI 目标主机的 CIDR 允许连接。default 将允许连接到任何 IPv4 地址。这个参数现在支持 IPv6 子网。设置 pure_iscsi_cidr_list 时忽略。
pure_iscsi_cidr_list = None	列表值	允许以逗号分隔的 FlashArray iSCSI 目标主机的 CIDR 列表。它支持 IPv4 和 IPv6 子网。此参数取代 pure_iscsi_cidr。
pure_replica_interval_default = 3600	整数值	快照复制间隔（以秒为单位）。
pure_replica_retention_logging_term_default = 7	整数值	目前，每天保留快照（以天数为单位）
pure_replica_retention_logging_term_per_day_default = 3	整数值	保留每天的快照数量。
pure_replica_retention_short_term_default = 14400	整数值	在目标上保留所有快照（以秒为单位）。
pure_replication_pg_name = cinder-group	字符串值	用于 async 复制的纯保护组名称（如果不存在，则会创建）。
pure_replication_pod_name = cinder-pod	字符串值	用于同步复制的纯 Pod 名称（如果不存在，则会创建）。
pvme_iscsi_ips = []	列表值	以逗号分隔的目标 iSCSI IP 地址列表。
pvme_pool_name = A	字符串值	用于创建卷的池或 Vdisk 名称。
qnap_management_url = None	URI 值	管理 QNAP 存储的 URL。驱动程序不支持 URL 中的 IPv6 地址。
qnap_poolname = None	字符串值	QNAP Storage 中的池名称
qnap_storage_protocol = iscsi	字符串值	用于访问 QNAP 存储的通信协议
quobyte_client_cfg = None	字符串值	指向 Quobyte 客户端配置文件的路径。
quobyte_mount_point_base = \$state_path/mnt	字符串值	包含 Quobyte 卷的挂载点的基本目录。

配置选项 = 默认值	类型	Description
quobyte_overlay_volumes = False	布尔值	通过创建覆盖文件而不是完整副本，从 volume_from_snapshot_cache 创建新卷。这可加快从这个缓存中创建卷。此功能要求选项 quobyte_qcow2_volumes 和 quobyte_volume_from_snapshot_cache 设置为 True。如果其中之一被设置为 False，则忽略此选项。
quobyte_qcow2_volumes = True	布尔值	将卷创建为 QCOW2 文件，而不是 raw 文件。
quobyte_sparsed_volumes = True	布尔值	将卷创建为稀疏文件，占用任何空间。如果设置为 False，则卷作为常规文件创建。
quobyte_volume_from_snapshot_cache = False	布尔值	从合并的快照创建卷缓存，以便加快从单个快照创建多个卷。
quobyte_volume_url = None	字符串值	使用 e.g. DNS SRV 记录（首选）或主机列表（如 quobyte://<DIR host1>, <DIR host2>/<volume name>）的 Quobyte volume 的 quobyte URL。
rados_connect_timeout = -1	整数值	连接到 ceph 集群时使用的超时值（以秒为单位）。如果值 < 0，则不会设置超时，并且使用默认的 librados 值。
rados_connection_interval = 5	整数值	连接重试 ceph 集群之间的间隔值（以秒为单位）。
rados_connection_retries = 3	整数值	如果连接到 ceph 集群的连接失败，重试次数。
<code>`rbd_ceph_conf = `</code>	字符串值	ceph 配置文件的路径
rbd_cluster_name = ceph	字符串值	ceph 集群的名称
rbd_exclusive_cinder_pool = True	布尔值	如果池与其他用法共享，则设置为 False。在专用使用驱动程序上，不会查询镜像的调配大小，因为它们将与用于 allocated_capacity_gb 的 Cinder 核心代码计算的匹配。这可减少 Ceph 集群以及卷服务的负载。在非专用驱动程序上，将为每个使用磁盘的每个镜像查询 Ceph 集群，这是为每个镜像具有独立请求的密集型操作。
rbd_flatten_volume_from_snapshot = False	布尔值	扁平化从快照创建的卷，将依赖项从卷删除到快照
rbd_iscsi_api_debug = False	布尔值	启用客户端请求调试。

配置选项 = 默认值	类型	Description
<code>`rd_iscsi_api_password = `</code>	字符串值	rd_target_api 服务的用户名
<code>`rd_iscsi_api_url = `</code>	字符串值	rd_target_api 服务的 url
<code>`rd_iscsi_api_user = `</code>	字符串值	rd_target_api 服务的用户名
rd_iscsi_target_iqn = None	字符串值	iscsi 网关上预配置的 target_iqn。
rd_max_clone_depth = 5	整数值	发生扁平化前的嵌套卷克隆的最大数量。设置为 0 以禁用克隆。注：降低这个值不会影响克隆深度超过新值的现有卷。
rd_pool = rbd	字符串值	存储 rbd 卷的 RADOS 池
rd_secret_uuid = None	字符串值	rd_user 卷的 secret 的 libvirt uuid
rd_store_chunk_size = 4	整数值	卷将阻断到这个大小的对象中（以 MB 为单位）。
rd_user = None	字符串值	用于访问 rbd 卷的 RADOS 客户端名称 - 仅在使用 cephx 身份验证时设置
remove_empty_host = False	布尔值	当最后一个 LUN 从中分离时，要从其中删除主机。默认情况下，它是 False。
replication_connect_timeout = 5	整数值	连接到 ceph 集群时使用的超时值（以秒为单位），以执行卷的降级/提升。如果值 < 0，则不会设置超时，并且使用默认的 librados 值。
replication_device = None	dict 值	多选择代表复制目标设备的字典。这个选项可以在单一配置部分中多次指定，以指定多个复制目标设备。每个条目都使用标准字典配置表单： replication_device = target_device_id: <required>,key1:value1,key2:value2...
report_discard_supported = False	布尔值	报告后端支持丢弃的 Cinder 客户端（也称为 trim/unmap）。这不会实际更改后端或客户端的行为，它只通知它可以被使用。
report_dynamic_total_capacity = True	布尔值	为 driver 设置为 True，将总容量报告为动态值（使用 + current free）和 False 以报告静态值（如果已定义且集群全局大小则为 quota max 字节）。
reserved_percentage = 0	整数值	后端容量的百分比被保留
retries = 200	整数值	使用这个值指定重试次数。

配置选项 = 默认值	类型	Description
san_api_port = None	端口值	用于访问 SAN API 的端口
<code>`san_clustername = `</code>	字符串值	用于创建卷的集群名称
<code>`san_ip = `</code>	字符串值	SAN 控制器的 IP 地址
san_is_local = False	布尔值	在本地而不是通过 SSH 执行命令；如果卷服务运行在 SAN 设备上，请使用。
san_login = admin	字符串值	SAN 控制器的用户名
<code>`san_password = `</code>	字符串值	SAN 控制器的密码
<code>`san_private_key = `</code>	字符串值	用于 SSH 验证的私钥的文件名
san_ssh_port = 22	端口值	用于 SAN 的 SSH 端口
san_thin_provision = True	布尔值	对 SAN 卷使用精简配置？
scst_target_driver = iscsi	字符串值	SCST 目标实施可以从多个 SCST 目标驱动程序中选择。
scst_target_iqn_name = None	字符串值	某些 iSCSI 目标具有预定义的目标名称，SCST 目标驱动程序使用此名称。
seagate_iscsi_ips = []	列表值	以逗号分隔的目标 iSCSI IP 地址列表。
seagate_pool_name = A	字符串值	用于卷创建的池或 vdisk 名称。
seagate_pool_type = virtual	字符串值	线性（用于 vdisk）或虚拟（用于虚拟池）。
<code>`secondary_san_ip = `</code>	字符串值	二级 DSM 控制器的 IP 地址
secondary_san_login = Admin	字符串值	二级 DSM 用户名
<code>`secondary_san_password = `</code>	字符串值	Secondary DSM 用户密码名称
secondary_sc_api_port = 3033	端口值	二级 Dell API 端口

配置选项 = 默认值	类型	Description
sf_account_prefix = None	字符串值	使用这个前缀创建 SolidFire 帐户。此处可以使用任何字符串，但字符串"hostname"是特殊的，将使用 cinder 节点主机名（之前的默认行为）创建一个前缀。默认值为 NO 前缀。
sf_allow_tenant_qos = False	布尔值	允许租户在创建时指定 QOS
sf_api_port = 443	端口值	SolidFire API 端口。如果设备 api 位于不同端口上的代理后面，则很有用。
sf_api_request_timeout = 30	整数值	设置等待 api 请求完成的时间（以秒为单位）。
sf_cluster_pairing_timeout = 60	整数值	设置等待集群完成对的时间（以秒为单位）。
sf_emulate_512 = True	布尔值	在创建卷时设置 512 字节模拟；
sf_enable_vag = False	布尔值	根据每个租户使用卷访问组。
sf_provisioning_calc = maxProvisionedSpace	字符串值	更改 SolidFire 报告已用空间和调配计算的方式。如果此参数设为 <i>usedSpace</i> ，则驱动程序将报告 Cinder 精简配置预期的正确值。
sf_svip = None	字符串值	使用指定的默认集群 SVIP 覆盖。这是实施在其云中 iSCSI 网络使用 VLAN 的必需部署。
sf_volume_clone_timeout = 600	整数值	设置等待卷或快照克隆完成的时间（以秒为单位）。
sf_volume_create_timeout = 60	整数值	设置等待创建卷操作完成的时间（以秒为单位）。
sf_volume_pairing_timeout = 3600	整数值	设置等待迁移卷完成对和同步的时间（以秒为单位）。
sf_volume_prefix = UUID-	字符串值	使用这个前缀创建 SolidFire 卷。卷名称的格式是 <sf_volume_prefix><cinder-volume-id>。默认为使用 <i>UUID-</i> 前缀。
smbfs_default_volume_format = vhd	字符串值	如果没有指定卷格式，则在创建卷时将使用的默认格式。
smbfs_mount_point_base = C:\OpenStack\mnt	字符串值	包含 smbfs 共享的挂载点的基本目录。

配置选项 = 默认值	类型	Description
smbfs_pool_mappings = {}	dict 值	共享位置和池名称之间的映射。如果未指定，共享名称将用作池名称。示例： //addr/share:pool_name, //addr/share2:pool_name2
smbfs_shares_config = C:\OpenStack\smbfs_shares.txt	字符串值	带有可用 smbfs 共享列表的文件。
spdk_max_queue_depth = 64	整数值	rdma 传输的队列深度。
spdk_rpc_ip = None	字符串值	NVMe 目标远程配置 IP 地址。
spdk_rpc_password = None	字符串值	NVMe 目标远程配置密码。
spdk_rpc_port = 8000	端口值	NVMe 目标远程配置端口。
spdk_rpc_protocol = http	字符串值	与 SPDK RPC 代理一起使用的协议
spdk_rpc_username = None	字符串值	NVMe 目标远程配置用户名。
ssh_conn_timeout = 30	整数值	SSH 连接超时（以秒为单位）
ssh_max_pool_conn = 5	整数值	池中的最大 ssh 连接
ssh_min_pool_conn = 1	整数值	池中的最小 ssh 连接
storage_protocol = iscsi	字符串值	在主机和存储后端之间传输数据的协议。
storage_vnx_authentication_type = global	字符串值	VNX 身份验证范围类型。默认情况下，值为 global。
storage_vnx_pool_names = None	列表值	要使用的存储池名称的逗号分隔列表。
storage_vnx_security_file_dir = None	字符串值	包含 VNX 安全文件的目录路径。确保首先生成了安全文件。
storpool_replication = 3	整数值	默认的 StorPool 链复制值。如果没有设置 storpool_template，则在创建没有指定类型的卷时使用。还用于计算统计中报告的明显可用空间。
storpool_template = None	字符串值	没有类型的卷的 StorPool 模板。

配置选项 = 默认值	类型	Description
storwize_peer_pool = None	字符串值	指定 hyperswap 卷的对等池名称，对等池必须存在于其他站点上。
storwize_portset = None	字符串值	指定在其中创建主机的 portset 的名称。
storwize_preferred_host_site = {}	dict 值	指定主机的站点信息。可以指定主机中使用的一个 WWPN 或多 WWPN。例如： storwize_preferred_host_site=site1:wwpn1,site2:wwpn2&wwpn3 或 storwize_host_site=site1:iqn1,site2:iqn2
storwize_san_secondary_ip = None	字符串值	指定在 san_ip 无效或无法访问时使用的辅助管理 IP 或主机名。
storwize_svc_allow_tenant_qos = False	布尔值	允许租户在创建时指定 QOS
storwize_svc_flashcopy_rate = 50	整数值	指定创建完整卷副本时要使用的 Storwize FlashCopy 复制率。默认值为 50，有效率为 1-150。
storwize_svc_flashcopy_timeout = 120	整数值	等待 FlashCopy 准备的最大秒数。
storwize_svc_iscsi_chap_enabled = True	布尔值	为 iSCSI 连接配置 CHAP 身份验证（默认：Enabled）
storwize_svc_mirror_pool = None	字符串值	指定存储镜像副本的池的名称。示例："pool2"
storwize_svc_multihostmap_enabled = True	布尔值	这个选项不再有任何影响。它已被弃用，并将在以后的版本中删除。
storwize_svc_multipath_enabled = False	布尔值	与多路径连接（仅限 FC；iSCSI 多路径由 Nova 控制）
storwize_svc_retain_aux_volume = False	布尔值	在删除主存储上的卷或将主卷从镜像(mirror)到非镜像(mirror)时，启用或禁用辅助存储上保留 aux 卷(mirror)。这个选项对 Spectrum Virtualize Family 有效。
storwize_svc_stretched_cluster_partner = None	字符串值	如果以扩展集群模式运行，请指定存储镜像副本的池的名称。例如："pool2"
storwize_svc_vol_autoexpand = True	布尔值	卷的存储系统 autoexpand 参数(True/False)

配置选项 = 默认值	类型	Description
storwize_svc_vol_compression = False	布尔值	卷的存储系统压缩选项
storwize_svc_vol_easytier = True	布尔值	为卷启用 Easy Tier
storwize_svc_vol_grainsize = 256	整数值	存储系统获取卷的大小参数(8/32/64/128/256)
storwize_svc_vol_iogroup = 0	字符串值	分配卷的 I/O 组。它可以是一个用逗号分隔的列表，在这种情况下，驱动程序将根据与 io_group 关联的最少卷数量选择 io_group。
storwize_svc_vol_nofmtdisk = False	布尔值	指定在创建过程中不格式化卷。
storwize_svc_vol_rsize = 2	整数值	用于卷的存储系统空间效率参数（百分比）
storwize_svc_vol_warning = 0	整数值	卷容量警告的存储系统阈值（百分比）
storwize_svc_volpool_name = ['volpool']	列表值	卷存储系统存储池的逗号分隔列表。
suppress_requests_ssl_warnings = False	布尔值	抑制请求库 SSL 证书警告。
synology_admin_port = 5000	端口值	用于 Synology 存储的管理端口。
synology_device_id = None	字符串值	如果启用了 OTP，用于跳过一次性密码检查 Synology 存储中的设备 ID。
synology_one_time_pass = None	字符串值	如果启用了 OTP，则管理员登录 Synology 存储的一次性密码。
<code>`synology_password = `</code>	字符串值	用于登录 Synology 存储的管理员密码。
<code>`synology_pool_name = `</code>	字符串值	用于创建 lun 的 Synology 存储上的卷。
synology_ssl_verify = True	布尔值	如果 \$driver_use_ssl 为 True，则进行证书验证
synology_username = admin	字符串值	Synology 存储的管理员。

配置选项 = 默认值	类型	Description
target_helper = tgtadm	字符串值	使用目标用户固定工具。tgtadm 为默认，使用 lioadm 作为 LIO iSCSI 支持、SCST 目标支持的 scstadmin、ietadm for iSCSI Enterprise Target、iscsictl 用于 Chelsio iSCSI 目标，nvmef 用于 NVMe-oF 支持，spdk-nvmeof 用于 SPDK NVMe-oF，或为测试虚拟虚拟。注：IET 驱动程序已弃用，并将在 V 发行版本中删除。
target_ip_address = \$my_ip	字符串值	iSCSI/NVMe-oF 守护进程侦听的 IP 地址
target_port = 3260	端口值	iSCSI/NVMe-oF 守护进程侦听的端口
target_prefix = iqn.2010-10.org.openstack:	字符串值	iSCSI/NVMe-oF 卷的前缀
target_protocol = iscsi	字符串值	决定新卷的目标协议，使用 tgtadm、lioadm 和 nvmef 目标帮助程序创建。要启用 RDMA，此参数应该使用值 "iser" 设置。支持的 iSCSI 协议值为 "iscsi" 和 "iser"，如果 nvmef target 设置为 "nvmef_rdma" 或 "nvmef_tcp"。
target_secondary_ip_addresses = []	列表值	iSCSI/NVMe-oF 守护进程的辅助 IP 地址列表
thres_avl_size_perc_start = 20	整数值	如果 NFS 共享的可用空间百分比低于此选项指定的值，则将清理 NFS 镜像缓存。
thres_avl_size_perc_stop = 60	整数值	当 NFS 共享中的可用空间百分比达到此选项指定的百分比时，该驱动程序将停止清除过去 M 分钟内未访问的 NFS 镜像缓存中的文件，其中 M 是 expiry_thres_minutes 配置选项的值。
trace_flags = None	列表值	控制将哪些 trace info 写入 DEBUG 日志级别的选项列表，以帮助开发人员。有效值为 method 和 api。
u4p_failover_autofailback = True	布尔值	如果驱动程序应该会在重新建立连接时自动恢复到 Unisphere 的主实例。
u4p_failover_backoff_factor = 1	整数值	在第二次尝试后，在尝试之间应用 backoff 因素（大多数错误都会由第二次尝试立即解决，且无延迟）。重试将休眠： $\{\text{backoff factor}\} * (2^{(\text{共重试} - 1)})$ 秒。
u4p_failover_retries = 3	整数值	每个连接应尝试的最大重试次数。请注意，这只适用于失败的 DNS 查找、套接字连接和连接超时，永远不会向服务器发出数据的请求。

配置选项 = 默认值	类型	Description
u4p_failover_target = None	dict 值	Unisphere 故障转移目标信息的字典。
u4p_failover_timeout = 20.0	整数值	在放弃前等待服务器发送数据的时长。
unique_fqdn_network = True	布尔值	我们的专用网络是否在每个启动器上都有唯一的 FQDN。例如，具有 QA 系统的网络通常有多个具有相同 FQDN 的服务器/VM。为 true 时，这将使用 FQDN 在 3PAR 上创建主机条目，如果为 false，它将使用反向的 IQN/WWNN。
unity_io_ports = []	列表值	要使用的 iSCSI 或 FC 端口的逗号分隔列表。每个端口可以是 Unix 样式的 glob 表达式。
unity_storage_pool_names = []	列表值	要使用的存储池名称的逗号分隔列表。
use_chap_auth = False	布尔值	为目标启用/禁用 CHAP 验证的选项。
use_multipath_for_image_xfer = False	布尔值	我们是否使用多路径(image)将多路径附加到卷并镜像到卷传输中，在 cinder 中附加/分离卷？这个参数需要为每个 backend 部分或 [backend_defaults] 部分中配置，作为所有后端的通用配置。
vmax_workload = None	字符串值	工作负载，最好将其设置为 pool_name 中的额外规格。
vmware_adapter_type = lsiLogic	字符串值	用于附加卷的默认适配器类型。
vmware_api_retry_count = 10	整数值	在连接相关问题时，必须重试 VMware vCenter 服务器 API 的次数。
vmware_ca_file = None	字符串值	用于验证 vCenter 服务器证书时使用的 CA 捆绑包文件。
vmware_cluster_name = None	多值	应创建卷的 vCenter 计算集群的名称。
vmware_connection_pool_size = 10	整数值	http 连接池中的最大连接数。
vmware_datastore_regex = None	字符串值	正则表达式模式，与创建后端卷的数据存储名称匹配。

配置选项 = 默认值	类型	Description
vmware_enable_volume_stats = False	布尔值	如果为 true，这将启用从后端获取卷统计。这在扩展方面存在潜在的性能问题。当 False 时，驱动程序不会收集有关后端的 ANY stats。
vmware_host_ip = None	字符串值	用于连接到 VMware vCenter 服务器的 IP 地址。
vmware_host_password = None	字符串值	用于使用 VMware vCenter 服务器进行身份验证的密码。
vmware_host_port = 443	端口值	用于连接到 VMware vCenter 服务器的端口号。
vmware_host_username = None	字符串值	用于使用 VMware vCenter 服务器进行身份验证的用户名。
vmware_host_version = None	字符串值	指定 VMware vCenter 服务器版本的可选字符串。驱动程序尝试从 VMware vCenter 服务器检索版本。只有在您要覆盖 vCenter 服务器版本时，才设置此配置。
vmware_image_transfer_timeout_secs = 7200	整数值	Cinder 和 Glance 之间的 VMDK 卷传输的超时时间（以秒为单位）。
vmware_insecure = False	布尔值	如果为 true，则不会验证 vCenter 服务器证书。如果为 false，则使用默认 CA 信任存储进行验证。如果设置了 "vmware_ca_file"，则忽略这个选项。
vmware_lazy_create = True	布尔值	如果为 true，则在没有源的情况下创建卷时，vCenter 服务器中的后端卷会被创建。当卷被附加、上传到镜像服务或备份过程中，会创建后端卷。
vmware_max_objects_retrieval = 100	整数值	每个批处理要检索的最大对象数量。查询结果将从服务器批量获得，而不是在一个 shot 中获得。服务器仍然可以将计数限制为小于配置的值。
vmware_snapshot_format = template	字符串值	vCenter 服务器中的卷快照格式。
vmware_storage_profile = None	多值	要监控的存储配置集的名称。仅在 vmware_enable_volume_stats 为 True 时使用。
vmware_task_poll_interval = 2.0	浮点值	轮询 VMware vCenter 服务器上调用的远程任务的时间间隔（以秒为单位）。
vmware_tmp_dir = /tmp	字符串值	在卷备份和恢复过程中存储虚拟磁盘的目录。

配置选项 = 默认值	类型	Description
vmware_volume_folder = Volumes	字符串值	包含 Cinder 卷的 vCenter 清单文件夹的名称。此文件夹将在 "OpenStack/<project_folder>" 下创建，其中 project_folder 格式为 "Project (<volume_project_id>)"。
vmware_wsdl_location = None	字符串值	可选的 LEVEL 服务 WSDL 位置，如 <a href="http://<server>/vimService.wsdl">http://<server>/vimService.wsdl 。可选的 over-ride 到默认位置用于程序错误的工作。
vnx_async_migrate = True	布尔值	在卷克隆和从快照创建过程中，始终使用异步迁移。如配置文档中所述，sync 迁移有一些限制。除了使用元数据外，客户还可以使用这个选项禁用 async 迁移。请注意，当两者都被设置时，元数据中的 async_migrate 都会覆盖这个选项。默认情况下，值为 True。
volume_backend_name = None	字符串值	给定驱动程序实现的后端名称
volume_clear = zero	字符串值	用于擦除旧卷的方法
volume_clear_ionice = None	字符串值	标记用于更改在删除后用于 0 个卷的进程的 i/o 优先级，例如 "-c3" 只用于闲置优先级。
volume_clear_size = 0	整数值	在旧卷开始时擦除大小（以 MiB 为单位）。1024 MiB 为 max. 0 especially all
volume_copy_blkio_cgroup_name = cinder-volume-copy	字符串值	用于限制卷复制带宽的 blkio cgroup 名称
volume_copy_bps_limit = 0	整数值	卷副本的带宽上限。0 jpeg unlimited
volume_dd_blocksize = 1M	字符串值	复制/清理卷时使用的默认块大小
volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver	字符串值	用于卷创建的驱动程序
volume_group = cinder-volumes	字符串值	包含导出的卷的 VG 的名称
volumes_dir = \$state_path/volumes	字符串值	卷配置文件存储目录

配置选项 = 默认值	类型	Description
vxflexos_allow_migration_during_rebuild = False	布尔值	重命名为 powerflex_allow_migration_during_rebuild。
vxflexos_allow_non_padded_volumes = False	布尔值	重命名为 powerflex_allow_non_padded_volumes。
vxflexos_max_over_subscription_ratio = 10.0	浮点值	重命名为 powerflex_max_over_subscription_ratio。
vxflexos_rest_server_port = 443	端口值	重命名为 powerflex_rest_server_port。
vxflexos_round_volume_capacity = True	布尔值	重命名为 powerflex_round_volume_capacity。
vxflexos_server_api_version = None	字符串值	重命名为 powerflex_server_api_version。
vxflexos_storage_pools = None	字符串值	重命名为 powerflex_storage_pools。
vxflexos_unmap_volume_before_deletion = False	布尔值	重命名为 powerflex_round_volume_capacity。
vzstorage_default_volume_format = raw	字符串值	如果没有指定卷格式，则在创建卷时将使用的默认格式。
vzstorage_mount_options = None	列表值	传递给 vzstorage 客户端的挂载选项。详情请查看 pstorage-mount 手册页部分。
vzstorage_mount_point_base = \$state_path/mnt	字符串值	包含 vzstorage 共享的挂载点的基本目录。
vzstorage_shares_config = /etc/cinder/vzstorage_shares	字符串值	带有可用 vzstorage 共享列表的文件。
vzstorage_sparsed_volumes = True	布尔值	将卷创建为稀疏文件，在使用原始格式时不使用空格而不是常规文件，在这种情况下，卷创建需要大量时间。
vzstorage_used_ratio = 0.95	浮点值	在不可将新卷分配给卷目的地之前，底层卷的 ACTUAL 使用百分比。
windows_iscsi_lun_path = C:\iSCSIVirtualDisks	字符串值	存储 VHD 支持的卷的路径

配置选项 = 默认值	类型	Description
xtremio_array_busy_retry_count = 5	整数值	如果数组忙碌的重试次数
xtremio_array_busy_retry_interval = 5	整数值	如果数组忙碌的重试间隔
xtremio_clean_unused_ig = False	布尔值	如果驱动程序删除在最后一次连接终止后没有卷的启动器组。因为现在的行为是保留 IG，所以我们默认为 False（在没有连接卷的情况下删除 IG）；在终止到最后一个卷的连接后将删除任何 IG。
<code>xtremio_cluster_name = `</code>	字符串值	多集群环境中的 XMS 集群 ID
xtremio_ports = []	列表值	允许的端口。以逗号分隔的 XtremIO iSCSI IP 或 FC WWN（ex.要使用的 58:CC:f0:98:49:22:07:02）。如果没有设置选项，则允许所有端口。
xtremio_volumes_per_glance_cache = 100	整数值	从每个缓存的 glance 镜像创建的卷数
zadara_access_key = None	字符串值	VPSA 访问密钥
zadara_default_snap_policy = False	布尔值	VPSA - 为卷附加快照策略。如果选项没有配置为元数据，则 VPSA 将继承默认值。
zadara_gen3_vol_compress = False	布尔值	VPSA - 为卷启用压缩。如果选项没有配置为元数据，则 VPSA 将继承默认值。
zadara_gen3_vol_dedupe = False	布尔值	VPSA - 为卷启用重复数据删除。如果选项没有配置为元数据，则 VPSA 将继承默认值。
zadara_ssl_cert_verify = True	布尔值	如果设置为 True，http 客户端将验证 VPSA 端点的 SSL 证书。
zadara_vol_encrypt = False	布尔值	VPSA - 卷的默认加密策略。如果选项没有配置为元数据，则 VPSA 将继承默认值。
zadara_vpsa_host = None	主机地址值	VPSA - 管理主机名或 IP 地址
zadara_vpsa_poolname = None	字符串值	VPSA - 为卷分配的存储池
zadara_vpsa_port = None	端口值	VPSA - 端口号

配置选项 = 默认值	类型	Description
zadara_vpsa_use_ssl = False	布尔值	VPSA - 使用 SSL 连接

2.1.4. barbican

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[barbican]` 组下可用的选项。

表 2.3. barbican

配置选项 = 默认值	类型	Description
auth_endpoint = http://localhost/identity/v3	字符串值	使用此端点连接到 Keystone
barbican_api_version = None	字符串值	Barbican API 的版本，例如：“v1”
barbican_endpoint = None	字符串值	使用此端点连接到 Barbican，例如：“http://localhost:9311/”
barbican_endpoint_type = public	字符串值	指定端点的类型。允许的值有：public、private 和 admin
number_of_retries = 60	整数值	在创建密钥时重试轮询的次数
retry_delay = 1	整数值	重试轮询创建密钥完成前等待的秒数
verify_ssl = True	布尔值	指定是否不安全的 TLS (https) 请求。如果为 False，则不会验证服务器证书，如果为 True，则可以同时设置 <code>verify_ssl_path</code> 配置。
verify_ssl_path = None	字符串值	要检查的捆绑包或 CA 证书的路径，或 None 请求尝试查找和使用 <code>verify_ssh</code> 为 True 的证书。如果 <code>verify_ssl</code> 是 False，这将被忽略。

2.1.5. brcd_fabric_example

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[brcd_fabric_example]` 组下可用的选项。

表 2.4. brcd_fabric_example

配置选项 = 默认值	类型	Description
<code>`fc_fabric_address = `</code>	字符串值	Fabric 管理 IP。
<code>`fc_fabric_password = `</code>	字符串值	用户的密码。

配置选项 = 默认值	类型	Description
fc_fabric_port = 22	端口值	连接端口
<code>`fc_fabric_ssh_cert_path = `</code>	字符串值	本地 SSH 证书路径。
<code>`fc_fabric_user = `</code>	字符串值	fabric 用户 ID。
fc_southbound_protocol = REST_HTTP	字符串值	光纤的南绑定连接器。
fc_virtual_fabric_id = None	字符串值	虚拟 Fabric ID。
zone_activate = True	布尔值	覆盖 zoning 激活状态。
zone_name_prefix = openstack	字符串值	覆盖区域名称前缀。
zoning_policy = initiator-target	字符串值	覆盖 zoning 策略。

2.1.6. cisco_fabric_example

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[cisco_fabric_example]` 组下可用的选项。

表 2.5. cisco_fabric_example

配置选项 = 默认值	类型	Description
<code>`cisco_fc_fabric_address = `</code>	字符串值	fabric 管理 IP
<code>`cisco_fc_fabric_password = `</code>	字符串值	用户的密码
cisco_fc_fabric_port = 22	端口值	连接端口
<code>`cisco_fc_fabric_user = `</code>	字符串值	Fabric 用户 ID
cisco_zone_activate = True	布尔值	覆盖 zoning 激活状态
cisco_zone_name_prefix = None	字符串值	覆盖区名称前缀
cisco_zoning_policy = initiator-target	字符串值	覆盖 zoning 策略

配置选项 = 默认值	类型	Description
cisco_zoning_vsan = None	字符串值	Fabric 的 VSAN

2.1.7. 协调

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[coordination]` 组下可用的选项。

表 2.6. 协调

配置选项 = 默认值	类型	Description
backend_url = file://\$state_path	字符串值	用于分布式协调的后端 URL。

2.1.8. CORS

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[cors]` 组下可用的选项。

表 2.7. CORS

配置选项 = 默认值	类型	Description
allow_credentials = True	布尔值	指明实际请求可以包含用户凭证
allow_headers = ['X-Auth-Token', 'X-Identity-Status', 'X-Roles', 'X-Service-Catalog', 'X-User-Id', 'X-Tenant-Id', 'X-OpenStack-Request-ID', 'X-Trace-Info', 'X-Trace-HMAC', 'OpenStack-API-Version']	列表值	指定在实际请求过程中可以使用哪些标头字段名称。
allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH', 'HEAD']	列表值	指明在实际请求期间可以使用哪些方法。
allowed_origin = None	列表值	指明此资源是否可以与请求"origin"标头中收到的域共享。格式： <code>"<protocol>://<host>[:<port>]"</code> ，没有尾部斜杠。示例： https://horizon.example.com

配置选项 = 默认值	类型	Description
expose_headers = ['X-Auth-Token', 'X-Subject-Token', 'X-Service-Token', 'X-OpenStack-Request-ID', 'OpenStack-API-Version']	列表值	指明哪些标头可以安全地公开给 API。默认为 HTTP Simple Headers。
max_age = 3600	整数值	CORS preflight 请求的最大缓存期限。

2.1.9. database

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[database]` 组下可用的选项。

表 2.8. database

配置选项 = 默认值	类型	Description
backend = sqlalchemy	字符串值	用于数据库的后端。
connection = None	字符串值	用于连接到数据库的 SQLAlchemy 连接字符串。
connection_debug = 0	整数值	SQL 调试信息的详细程度：0=None, 100=Everything。
<code>connection_parameters = `</code>	字符串值	用于连接时附加到连接 URL 的可选 URL 参数；指定 <code>param1=value1&param2=value2&...</code>
connection_recycle_time = 3600	整数值	连接池中存在的连接的时间超过这个秒数，下一次从池中签出它们的时间将替换为新的连接。
connection_trace = False	布尔值	将 Python 堆栈跟踪作为注释字符串添加到 SQL。
db_inc_retry_interval = True	布尔值	如果为 True，将数据库操作的重试间隔增加到 <code>db_max_retry_interval</code> 。
db_max_retries = 20	整数值	在引发错误前出现连接错误或死锁错误时的最大重试次数。设置为 -1 以指定无限重试计数。
db_max_retry_interval = 10	整数值	如果设置了 <code>db_inc_retry_interval</code> ，则数据库操作重试之间的最大秒数。
db_retry_interval = 1	整数值	数据库事务重试之间的秒数。
max_overflow = 50	整数值	如果设置，请将这个值用于 SQLAlchemy 的 <code>max_overflow</code> 。

配置选项 = 默认值	类型	Description
max_pool_size = 5	整数值	在池中保留的最大 SQL 连接数。将值设为 0 表示没有限制。
max_retries = 10	整数值	启动期间数据库连接重试的最大数量。设置为 -1 以指定无限重试计数。
mysql_enable_ndb = False	布尔值	如果为 True，则透明地启用对处理 MySQL 集群 (NDB) 的支持。
mysql_sql_mode = TRADITIONAL	字符串值	用于 MySQL 会话的 SQL 模式。这个选项（包括默认选项）会覆盖任何 server-set SQL 模式。要使用服务器配置设置的任何 SQL 模式，请将其设置为 no 值。示例：mysql_sql_mode=
pool_timeout = None	整数值	如果设置，请将这个值用于 SQLAlchemy 的 pool_timeout。
retry_interval = 10	整数值	重新尝试打开 SQL 连接之间的间隔。
slave_connection = None	字符串值	用于连接到从数据库的 SQLAlchemy 连接字符串。
sqlite_synchronous = True	布尔值	如果为 True，则 SQLite 使用同步模式。
use_db_reconnect = False	布尔值	启用在连接丢失时进行数据库重新连接的实验性使用。

2.1.10. fc-zone-manager

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[fc-zone-manager]` 组下可用的选项。

表 2.9. fc-zone-manager

配置选项 = 默认值	类型	Description
brcd_sb_connector = HTTP	字符串值	用于 zoning 操作的南绑定连接器
cisco_sb_connector = cinder.zonemanager.drivers.cisco.cisco_fc_zone_client_cli.CiscoFCZoneClientCLI	字符串值	zoning 操作的南向连接器

配置选项 = 默认值	类型	Description
enable_unsupported_driver = False	布尔值	当您要允许不支持的区管理器驱动程序启动时，将其设置为 True。没有维护正常工作的 CI 系统和测试的驱动程序被标记为不受支持，直到 CI 再次工作为止。这也会将驱动程序标记为已弃用，并可在下一发行版本中删除。
fc_fabric_names = None	字符串值	以逗号分隔的光纤通道结构名称列表。此名称列表用于检索其他 SAN 凭据以连接到每个 SAN 光纤
fc_san_lookup_service = cinder.zonemanager.drivers.brocade.brcd_fc_san_lookup_service.BrcdFCSanLookupService	字符串值	FC SAN Lookup 服务
zone_driver = cinder.zonemanager.drivers.brocade.brcd_fc_zone_driver.BrcdFCZoneDriver	字符串值	负责区域管理的 FC 区域驱动程序
zoning_policy = initiator-target	字符串值	zoning 策略由用户配置；有效值包括"initiator-target"或"initiator"

2.1.11. healthcheck

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[healthcheck]` 组下可用的选项。

表 2.10. healthcheck

配置选项 = 默认值	类型	Description
backends = []	列表值	可执行健康检查并报告该信息作为请求的一部分的其他后端。
detailed = False	布尔值	显示响应中更详细的信息。安全备注：启用此选项可能会公开有关被监控服务的敏感详情。务必验证它是否不会违反您的安全策略。
disable_by_file_path = None	字符串值	检查是否存在文件，以确定应用是否在端口上运行。DisableByFileHealthcheck 插件使用。
disable_by_file_paths = []	列表值	根据端口检查文件是否存在，以确定应用是否在端口上运行。需要字符串的 "port:path" 列表。DisableByFilesPortsHealthcheck 插件使用。
path = /healthcheck	字符串值	响应 healthcheck 请求的路径。

2.1.12. key_manager

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[key_manager]` 组下可用的选项。

表 2.11. key_manager

配置选项 = 默认值	类型	Description
auth_type = None	字符串值	要创建的身份验证凭证的类型。可能的值有 <code>令牌</code> 、 <code>密码</code> 、 <code>keystone_token</code> 和 <code>keystone_password</code> 。如果没有将上下文传递给凭证工厂，则需要此项。
auth_url = None	字符串值	使用此端点连接到 Keystone。
backend = barbican	字符串值	指定密钥管理器实现。选项为 "barbican" 和 "vault"。默认为 "barbican"。将支持之前使用 <code>[key_manager]/api_class</code> 来设置的值。
domain_id = None	字符串值	域范围的域 ID。 <code>keystone_token</code> 和 <code>keystone_password</code> <code>auth_type</code> 是可选的。
domain_name = None	字符串值	域范围的域名。 <code>keystone_token</code> 和 <code>keystone_password</code> <code>auth_type</code> 是可选的。
fixed_key = None	字符串值	修复了密钥管理器返回的密钥，在 hex 中指定
password = None	字符串值	用于身份验证的密码。 <code>password</code> 和 <code>keystone_password</code> <code>auth_type</code> 需要。
project_domain_id = None	字符串值	项目的项目域 ID。 <code>keystone_token</code> 和 <code>keystone_password</code> <code>auth_type</code> 是可选的。
project_domain_name = None	字符串值	项目的项目的域名。 <code>keystone_token</code> 和 <code>keystone_password</code> <code>auth_type</code> 是可选的。
project_id = None	字符串值	项目范围的项目 ID。 <code>keystone_token</code> 和 <code>keystone_password</code> <code>auth_type</code> 是可选的。
project_name = None	字符串值	项目范围的项目名称。 <code>keystone_token</code> 和 <code>keystone_password</code> <code>auth_type</code> 是可选的。
reauthenticate = True	布尔值	如果当前令牌即将过期，则允许获取新令牌。 <code>keystone_token</code> 和 <code>keystone_password</code> <code>auth_type</code> 是可选的。
token = None	字符串值	进行身份验证的令牌。如果没有将上下文传递给凭证工厂， <code>token</code> 和 <code>keystone_token</code> <code>auth_type</code> 需要。
trust_id = None	字符串值	信任范围的信任 ID。 <code>keystone_token</code> 和 <code>keystone_password</code> <code>auth_type</code> 是可选的。

配置选项 = 默认值	类型	Description
user_domain_id = None	字符串值	用于身份验证的用户域 ID。 <i>keystone_token</i> 和 <i>keystone_password</i> <i>auth_type</i> 是可选的。
user_domain_name = None	字符串值	用于身份验证的用户域名。 <i>keystone_token</i> 和 <i>keystone_password</i> <i>auth_type</i> 是可选的。
user_id = None	字符串值	用于身份验证的用户 ID。 <i>keystone_token</i> 和 <i>keystone_password</i> <i>auth_type</i> 是可选的。
username = None	字符串值	进行身份验证的用户名。 <i>password</i> <i>auth_type</i> 需要。 <i>keystone_password</i> <i>auth_type</i> 是可选的。

2.1.13. keystone_auth token

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[keystone_auth token]` 组下可用的选项。

表 2.12. keystone_auth token

配置选项 = 默认值	类型	Description
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
auth_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。这个选项已弃用，现在使用 <code>www_authenticate_uri</code> ，并将在 S 发行版本中删除。弃用自：Queens *reason:* <code>auth_uri</code> 选项已弃用，现在使用 <code>www_authenticate_uri</code> ，并将在 S 发行版本中删除。
auth_version = None	字符串值	Identity API 端点的 API 版本。
cache = None	字符串值	存储 Swift 缓存对象的请求环境密钥。当使用 Swift 缓存部署 <code>auth_token</code> 中间件时，请使用此选项使中间件共享具有 <code>swift</code> 的缓存后端。否则，使用 <code>memcached_servers</code> 选项。
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。默认为系统 CA。

配置选项 = 默认值	类型	Description
certFile = None	字符串值	如果身份服务器需要客户端证书，则需要此项
delay_auth_decision = False	布尔值	不要在中间件内处理授权请求，而是将授权决策委派给下游 WSGI 组件。
enforce_token_bind = permissive	字符串值	用于控制令牌绑定的使用和类型。可以设置为 "disabled" 以不检查令牌绑定。如果绑定类型是已知服务器的绑定类型，则验证绑定信息，并忽略它（如 "permissive"），但如果绑定类型未知，则令牌将被拒绝。"必需"令牌绑定需要允许任何形式的令牌绑定。最后，令牌中必须存在的绑定方法名称。
http_connect_timeout = None	整数值	请求超时值与身份 API 服务器通信。
http_request_max_retries = 3	整数值	与身份 API 服务器通信时，我们尝试重新连接的次数。
include_service_catalog = True	布尔值	（可选）指示是否设置 X-Service-Catalog 标头。如果为 False，则中间件不会要求在令牌验证时请求服务目录，且不会设置 X-Service-Catalog 标头。
insecure = False	布尔值	验证 HTTPS 连接。
interface = internal	字符串值	用于 Identity API 端点的接口。有效值为 "public", "internal"（默认）或 "admin"。
keyfile = None	字符串值	如果身份服务器需要客户端证书，则需要此项
memcache_pool_conn_get_timeout = 10	整数值	（可选）操作等待从池中获取 memcached 客户端连接的秒数。
memcache_pool_dead_retry = 300	整数值	（可选）在再次尝试前 memcached 服务器被视为 dead 的秒数。
memcache_pool_maxsize = 10	整数值	（可选）到每个 memcached 服务器的最大打开连接数。
memcache_pool_socket_timeout = 3	整数值	（可选）与 memcached 服务器通信的套接字超时（以秒为单位）。
memcache_pool_unused_timeout = 60	整数值	（可选）与 memcached 的连接在池关闭前未使用的秒数。
memcache_secret_key = None	字符串值	（可选，如果定义了 memcache_security_strategy）此字符串用于密钥派生。

配置选项 = 默认值	类型	Description
memcache_security_strategy = None	字符串值	(可选) 如果定义, 则指示令牌数据是否应进行身份验证或加密。如果 MAC, 令牌数据在缓存中进行身份验证 (使用 HMAC)。如果 ENCRYPT, 令牌数据会在缓存中加密并进行身份验证。如果该值不是这些选项之一或为空, 则 auth_token 会在初始化时引发异常。
memcache_use_advanced_pool = False	布尔值	(可选) 使用 advanced (事件安全) memcached 客户端池。高级池只能在 python 2.x 下工作。
memcached_servers = None	列表值	(可选) 指定要用于缓存的 memcached 服务器列表。如果未定义, 令牌将会被缓存在进程中。
region_name = None	字符串值	找到身份服务器的区域。
service_token_roles = ['service']	列表值	需要出现在服务令牌中的角色。服务令牌可以请求可以使用过期的令牌, 因此此检查应严格控制只有实际服务应发送此令牌。此处的角色作为 ANY 检查应用, 因此此列表中的任何角色都必须存在。为了向后兼容的原因, 这目前只会影响 allow_expired 检查。
service_token_roles_required = False	布尔值	为了向后兼容的原因, 我们必须允许有效的服务令牌传递, 但不允许将 service_token_roles 检查视为有效。在以后的发行版本中设置此 true 将成为默认设置, 并应启用 (如果可能)。
service_type = None	字符串值	服务的名称或类型 (在服务目录中出现)。这用于验证具有受限访问规则的令牌。
token_cache_time = 300	整数值	为了防止过度努力验证令牌, 中间件会缓存之前可预见的令牌 (以秒为单位)。设置为 -1 可完全禁用缓存。
www_authenticate_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点, 因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下, 这个端点应该被未指定版本, 但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点, 则该服务用户不应该用于验证令牌的端点, 因为普通用户可能无法访问该端点。

2.1.14. nova

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[nova]` 组下可用的选项。

表 2.13. nova

配置选项 = 默认值	类型	Description
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
insecure = False	布尔值	验证 HTTPS 连接。
interface = public	字符串值	要使用的 nova 端点的类型。此端点将在 keystone 目录中查找，并且应该是公共、内部或 admin 之一。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
region_name = None	字符串值	要使用的 nova 区域名称。如果 keystone 管理多个区域，则很有用。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
timeout = None	整数值	http 请求的超时值
token_auth_url = None	字符串值	使用当前用户令牌时 nova 连接的身份验证 URL

2.1.15. oslo_concurrency

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[oslo_concurrency]` 组下可用的选项。

表 2.14. oslo_concurrency

配置选项 = 默认值	类型	Description
disable_process_locking = False	布尔值	启用或禁用进程间锁定。
lock_path = None	字符串值	用于锁定文件的目录。为安全起见，指定目录应只由运行需要锁定的进程的用户写入。默认为环境变量 <code>OSLO_LOCK_PATH</code> 。如果使用外部锁定，则必须设置锁定路径。

2.1.16. oslo_messaging_amqp

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[oslo_messaging_amqp]` 组下可用的选项。

表 2.15. oslo_messaging_amqp

配置选项 = 默认值	类型	Description
addressing_mode = dynamic	字符串值	表示驱动程序使用的寻址模式。允许的值： <i>legacy</i> - 使用旧的不可路由寻址； <i>routable</i> - 使用可路由地址； <i>dynamic</i> - 如果消息总线不支持路由，使用 <i>legacy</i> ，如果支持，使用 <i>routable</i> 。
anycast_address = anycast	字符串值	在发送到一组消费者时附加到地址前缀。消息总线用来识别应在消费者之间以轮循方式发送的消息。
broadcast_prefix = broadcast	字符串值	广播到所有服务器时使用的地址前缀
connection_retry_backoff = 2	整数值	每次失败故障转移尝试后，将 <code>connection_retry_interval</code> 增加到这秒数。
connection_retry_interval = 1	整数值	尝试重新连接前暂停的秒数。
connection_retry_interval_max = 30	整数值	<code>connection_retry_interval</code> + <code>connection_retry_backoff</code> 的最大限制
container_name = None	字符串值	AMQP 容器的名称必须全局唯一。默认为生成的 UUID
default_notification_exchange = None	字符串值	通知地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_notification_exchange if set else control_exchange if set else <i>notify</i>
default_notify_timeout = 30	整数值	发送通知消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_reply_retry = 0	整数值	重新发送回复消息的最大尝试次数，因为可恢复错误而失败。
default_reply_timeout = 30	整数值	rpc 回复消息发送的截止时间。
default_rpc_exchange = None	字符串值	RPC 地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_rpc_exchange if set other control_exchange if set else <i>rpc</i>
default_send_timeout = 30	整数值	rpc cast 或调用消息发送的截止时间。仅在调用者不提供超时到期时使用。

配置选项 = 默认值	类型	Description
default_sender_link_timeout = 600	整数值	调度闲置发送者链接清除的持续时间。过期后分离链接。
group_request_prefix = unicast	字符串值	发送到组中任何服务器时的地址前缀
idle_timeout = 0	整数值	不活跃连接的超时时间（以秒为单位）
link_retry_delay = 10	整数值	重新连接 AMQP 1.0 链接之间的暂停时间，该链接因为可恢复的错误而失败。
multicast_address = multicast	字符串值	在发送异常消息时，附加到地址前缀。消息总线用来识别 fanout 消息。
notify_address_prefix = openstack.org/om/notify	字符串值	所有生成的通知地址的地址前缀
notify_server_credit = 100	整数值	传入通知消息的窗口大小
pre_settled = ['rpc-cast', 'rpc-reply']	多值	发送此类型的消息预先组装。预先设置的消息不会接收来自 peer 的确认信息。请注意：如果发送失败，则预先设置的消息可能会被静默丢弃。允许的值： <i>rpc-call</i> - 发送 RPC Calls <i>pre-settled</i> <i>rpc-reply</i> - 发送 RPC Replies <i>pre-settled</i> <i>rpc-cast</i> - Send RPC Casts <i>pre-settled</i> <i>notify</i> - Send Notifications <i>pre-settled</i>
pseudo_vhost = True	布尔值	为不支持虚拟主机的消息总线启用虚拟主机支持（如 qpid）。当设置为 true 时，虚拟主机名将添加到所有消息总线地址中，从而有效地为每个虚拟主机 <i>创建专用子网</i> 。如果消息总线支持使用 AMQP 1.0 Open performative 中的 <i>hostname</i> 字段作为虚拟主机的名称，设置为 False。
reply_link_credit = 200	整数值	传入 RPC 消息的窗口大小。
rpc_address_prefix = openstack.org/om/rpc	字符串值	所有生成的 RPC 地址的地址前缀
rpc_server_credit = 100	整数值	传入 RPC Request 消息的窗口大小
<code>`sas_config_dir = `</code>	字符串值	包含 SASL 配置的目录路径
<code>`sas_config_name = `</code>	字符串值	配置文件名称（不带 .conf 后缀）
<code>`sas_default_realm = `</code>	字符串值	如果用户名中没有 realm，则要使用的 SASL 域
<code>`sas_mechanisms = `</code>	字符串值	可接受的 SASL 机制的以空格分隔的列表

配置选项 = 默认值	类型	Description
server_request_prefix = exclusive	字符串值	发送到特定服务器时使用的地址前缀
ssl = False	布尔值	尝试通过 SSL 连接。如果没有给出其他与 ssl 相关的参数，它将使用系统的 CA-bundle 来验证服务器证书。
<code>`ssl_ca_file = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_cert_file = `</code>	字符串值	自我标识用于客户端身份验证的证书 PEM 文件
<code>`ssl_key_file = `</code>	字符串值	用于为 ssl_cert_file 证书签名的私钥 PEM 文件（可选）
ssl_key_password = None	字符串值	用于解密 ssl_key_file 的密码（如果加密）
ssl_verify_vhost = False	布尔值	默认情况下，SSL 检查服务器证书中的名称与 transport_url 中的主机名匹配。在某些配置中，最好使用虚拟主机名。例如，如果服务器使用 Server Name Indication TLS extension (rfc6066) 来为每个虚拟主机提供证书。如果服务器的 SSL 证书使用虚拟主机名称而不是 DNS 名称，则将 ssl_verify_vhost 设置为 True。
trace = False	布尔值	debug : 将 AMQP 帧转储到 stdout
unicast_address = unicast	字符串值	在发送到特定 RPC/Notification 服务器时，附加到地址前缀。消息总线用来识别发送到单个目的地的消息。

2.1.17. oslo_messaging_kafka

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[oslo_messaging_kafka]` 组下可用的选项。

表 2.16. oslo_messaging_kafka

配置选项 = 默认值	类型	Description
compression_codec = none	字符串值	producer 生成的所有数据的压缩 codec。如果没有设置，则不会使用压缩。请注意，这允许的值取决于 kafka 版本
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制

配置选项 = 默认值	类型	Description
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
consumer_group = oslo_messaging_consumer	字符串值	Kafka 消费者的组 ID。一个组中的消费者将协调消息消耗
enable_auto_commit = False	布尔值	启用异步消费者提交
kafka_consumer_timeout = 1.0	浮点值	Kafka 用户的默认超时
kafka_max_fetch_bytes = 1048576	整数值	Kafka consumer 的最大获取字节数
max_poll_records = 500	整数值	轮询调用中返回的最大记录数
pool_size = 10	整数值	Kafka Consumers 的池大小
producer_batch_size = 16384	整数值	producer async 发送的批处理大小
producer_batch_timeout = 0.0	浮点值	以秒为单位的 KafkaProducer 批处理延迟上限
sasl_mechanism = PLAIN	字符串值	当安全协议是 SASL 时的机制
security_protocol = PLAINTEXT	字符串值	用于与代理通信的协议
<code>`ssl_cafile = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_client_cert_file = `</code>	字符串值	用于身份验证的客户端证书 PEM 文件。
<code>`ssl_client_key_file = `</code>	字符串值	用于身份验证的客户端密钥 PEM 文件。
<code>`ssl_client_key_password = `</code>	字符串值	用于身份验证的客户端密钥密码文件。

2.1.18. oslo_messaging_notifications

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[oslo_messaging_notifications]` 组下可用的选项。

表 2.17. oslo_messaging_notifications

配置选项 = 默认值	类型	Description
driver = []	多值	处理发送通知的驱动程序。可能的值有 messaging, messagingv2, routing, log, test, noop
retry = -1	整数值	重新发送因为可恢复错误而发送失败的通知消息的最大尝试次数。0 - 不重试, -1 - indefinite
topics = ['notifications']	列表值	用于 OpenStack 通知的 AMQP 主题。
transport_url = None	字符串值	代表用于通知的消息传递驱动程序的 URL。如果没有设置, 我们回退到用于 RPC 的同一配置。

2.1.19. oslo_messaging_rabbit

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[oslo_messaging_rabbit]` 组下可用的选项。

表 2.18. oslo_messaging_rabbit

配置选项 = 默认值	类型	Description
amqp_auto_delete = False	布尔值	AMQP 中的自动删除队列。
amqp_durable_queues = False	布尔值	在 AMQP 中使用持久队列。
direct_mandatory_flag = True	布尔值	(DEPRECATED)为直接发送启用/禁用 RabbitMQ 强制标记。直接发送用作回复, 因此在客户端队列不存在时引发 MessageUndeliverable 异常。MessageUndeliverable 异常将用于循环使发送者恢复。此标志已弃用, 且无法停用此功能
enable_cancel_on_failover = False	布尔值	启用 x-cancel-on-ha-failover 标志, 使 rabbitmq 服务器将取消并通知 consumerswhen 队列停机
heartbeat_in_pthread = False	布尔值	默认情况下, 通过原生 python 线程运行健康检查心跳线程。如果此选项等于 False, 则健康检查 heartbeat 将继承父进程的执行模型。例如, 如果父进程使用 eventlet/greenlet 对 stdlib 进行补丁, 则心跳将通过绿色线程运行。这个选项应该只针对 wsgi 服务设置为 True。
heartbeat_rate = 2	整数值	heartbeat_timeout_threshold 期间的我们检查心跳的频率。
heartbeat_timeout_threshold = 60	整数值	如果 heartbeat 的 keep-alive 失败(0 禁用 heartbeat), 则 Rabbit 代理被视为 down 的秒数。

配置选项 = 默认值	类型	Description
kombu_compression = None	字符串值	EXPERIMENTAL : Possible 值为 : gzip、bz2。如果没有设置压缩, 则不会使用。这个选项可能在以后的版本中不可用。
kombu_failover_strategy = round-robin	字符串值	确定当当前连接到的 RabbitMQ 节点不可用时如何选择下一个 RabbitMQ 节点。仅在配置中提供多个 RabbitMQ 节点时生效。
kombu_missing_consumer_retry_timeout = 60	整数值	在放弃发送其回复前等待缺少的客户端的时间。这个值不应比 <code>rpc_response_timeout</code> 长。
kombu_reconnect_delay = 1.0	浮点值	响应 AMQP 消费者取消通知前等待多久。
rabbit_ha_queues = False	布尔值	尝试在 RabbitMQ 中使用 HA 队列(x-ha-policy: all)。如果更改这个选项, 您必须擦除 RabbitMQ 数据库。在 RabbitMQ 3.0 中, 在声明队列时, x-ha-policy 参数不再控制队列镜像。如果您只想确保所有队列(带有自动生成的名称的队列除外)都在所有节点上进行镜像, 请运行: <code>"rabbitmqctl set_policy HA ^(?!amq\.) gem {"ha-mode": "all"}"</code>
rabbit_interval_max = 30	整数值	RabbitMQ 连接重试的最大间隔。默认值为 30 秒。
rabbit_login_method = AMQPLAIN	字符串值	RabbitMQ 登录方法。
rabbit_qos_prefetch_count = 0	整数值	指定要预抓取的消息数量。设置为零可允许无限消息。
rabbit_retry_backoff = 2	整数值	连接到 RabbitMQ 时重试之间的 backoff 时长。
rabbit_retry_interval = 1	整数值	使用 RabbitMQ 重试连接的频率。
rabbit_transient_queues_ttl = 1800	整数值	正整数代表队列 TTL (x-expires)的持续时间(以秒为单位)。TTL 期间未使用的队列将被自动删除。参数仅影响 reply 和 fanout 队列。
ssl = False	布尔值	通过 SSL 连接。
<code>`ssl_ca_file = `</code>	字符串值	SSL 证书颁发机构文件(仅在启用了 SSL 时评估)。
<code>`ssl_cert_file = `</code>	字符串值	SSL 证书文件(仅在启用了 SSL 时评估)。
<code>`ssl_key_file = `</code>	字符串值	SSL 密钥文件(仅在启用了 SSL 时 valid)。

配置选项 = 默认值	类型	Description
<code>`ssl_version = `</code>	字符串值	要使用的 SSL 版本（仅在启用了 SSL 时评估）。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1_1 和 TLSv1_2 可能在某些发行版中提供。

2.1.20. oslo_middlewares

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[oslo_middlewares]` 组下可用的选项。

表 2.19. oslo_middlewares

配置选项 = 默认值	类型	Description
<code>enable_proxy_headers_parsing = False</code>	布尔值	应用程序是否在代理后面。这决定了中间件是否应该解析标头。
<code>max_request_body_size = 114688</code>	整数值	每个请求的最大正文大小，以字节为单位。
<code>secure_proxy_ssl_header = X-Forwarded-Proto</code>	字符串值	用于确定原始请求协议方案的 HTTP 标头，即使它被 SSL 终止代理隐藏也是如此。

2.1.21. oslo_policy

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[oslo_policy]` 组下可用的选项。

表 2.20. oslo_policy

配置选项 = 默认值	类型	描述
<code>enforce_new_defaults = False</code>	布尔值	这个选项控制在评估策略时是否使用旧的弃用默认值。如果为 True ，则不会评估旧的已弃用的默认值。这意味着，如果旧默认值允许任何现有令牌，但不允许使用新默认值，它将被禁止。我们鼓励启用此标志以及 <code>enforce_scope</code> 标志，以便您可以同时获得新默认值和 <code>scope_type</code> 的好处。
<code>enforce_scope = False</code>	布尔值	此选项控制在评估策略时是否强制执行范围。如果为 True ，则请求中使用的令牌范围将与被强制执行策略的 <code>scope_types</code> 进行比较。如果范围不匹配，则引发 <code>InvalidScope</code> 异常。如果为 False ，则会记录一条信息，告知操作员使用了不匹配的范围调用策略。
<code>policy_default_rule = default</code>	字符串值	默认规则。未找到请求的规则时强制执行。

配置选项 = 默认值	类型	描述
policy_dirs = ['policy.d']	多值	存储策略配置文件的目录。它们可以相对于 config_dir 选项或绝对路径定义的搜索路径中的任何目录。由 policy_file 定义的文件必须存在，才能搜索这些目录。缺少或空目录将被忽略。
policy_file = policy.yaml	字符串值	将角色映射到给定服务权限的文件的相对或绝对路径。必须使用与配置文件设置相关的相对路径。
remote_content_type = application/x-www-form-urlencoded	字符串值	用于发送和接收基于 REST 策略检查的数据的内容类型
remote_ssl_ca_cert_file = None	字符串值	基于 REST 的策略检查的 ca cert 文件的绝对路径
remote_ssl_client_cert_file = None	字符串值	基于 REST 的策略检查的客户端证书的绝对路径
remote_ssl_client_key_file = None	字符串值	绝对路径客户端密钥文件 REST 基于策略检查
remote_ssl_verify_server_cert = False	布尔值	基于 REST 的策略检查的服务器身份验证

2.1.22. oslo_reports

下表概述了 /etc/cinder/cinder.conf 文件中的 [oslo_reports] 组下可用的选项。

表 2.21. oslo_reports

配置选项 = 默认值	类型	Description
file_event_handler = None	字符串值	要监视触发报告更改的文件路径，而不是信号。设置此选项会禁用报告的信号触发器。如果应用程序作为 WSGI 应用程序运行，则建议使用此选项而不是信号。
file_event_handler_interval = 1	整数值	设置 file_event_handler 时轮询之间等待的秒数
log_dir = None	字符串值	创建文件的路径

2.1.23. oslo_versionedobjects

下表概述了 /etc/cinder/cinder.conf 文件中的 [oslo_versionedobjects] 组下可用的选项。

表 2.22. oslo_versionedobjects

配置选项 = 默认值	类型	Description
fatal_exception_format_errors = False	布尔值	使异常消息格式错误致命

2.1.24. privsep

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[privsep]` 组下可用的选项。

表 2.23. privsep

配置选项 = 默认值	类型	Description
capabilities = []	列表值	privsep 守护进程保留的 Linux 功能列表。
group = None	字符串值	privsep 守护进程应作为。
helper_command = None	字符串值	如果不使用"fork"方法，调用以启动 privsep 守护进程。如果没有指定，则使用"sudo privsep-helper"和用于重新创建当前配置参数生成默认值。此命令必须接受合适的 <code>--privsep_context</code> 和 <code>--privsep_sock_path</code> 参数。
logger_name = oslo_privsep.daemon	字符串值	用于此 privsep 上下文的日志记录器名称。默认情况下，带有 <code>oslo_privsep.daemon</code> 的所有上下文日志。
thread_pool_size = <based on operating system>	整数值	可用于 privsep 的线程数量，以同时运行进程。默认为系统中的 CPU 内核数。
user = None	字符串值	privsep 守护进程应以。

2.1.25. profiler

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[profiler]` 组下可用的选项。

表 2.24. profiler

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
connection_string = messaging://	字符串值	<p>通知程序后端的连接字符串。</p> <p>默认值为 messaging://，它将通知程序设置为 oslo_messaging。</p> <p>可能的值示例：</p> <ul style="list-style-type: none"> ● messaging:// - 使用 oslo_messaging 驱动程序来发送范围。 ● redis://127.0.0.1:6379 - 使用 redis 驱动程序发送 span。 ● MongoDB://127.0.0.1:27017 - 使用 mongodb 驱动程序进行发送范围。 ● Elasticsearch://127.0.0.1:9200 - 使用 elasticsearch 驱动程序来发送 span。 ● Jaeger://127.0.0.1:6831 - 使用 jaeger tracing 作为发送 span 的驱动程序。
enabled = False	布尔值	<p>为此节点上的所有服务启用性能分析。</p> <p>默认值为 False（完全禁用性能分析功能）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用该功能 ● false：禁用该功能。分析无法通过此项目操作启动。如果性能分析由另一个项目触发，则此项目部分将为空。
es_doc_type = notification	字符串值	elasticsearch 中通知索引的文档类型。
es_scroll_size = 10000	整数值	Elasticsearch 批量分割大型请求。此参数定义每个批处理的最大大小（例如：es_scroll_size=10000）。
es_scroll_time = 2m	字符串值	此参数是一个时间值参数（例如：es_scroll_time=2m），指示参与搜索的节点将维护相关资源以继续和支持它。

配置选项 = 默认值	类型	Description
filter_error_trace = False	布尔值	<p>启用包含 error/exception 的过滤器追踪来分隔的位置。</p> <p>默认值为 False。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用包含 error/exception 的过滤器追踪。 ● false：禁用过滤器。
hmac_keys = SECRET_KEY	字符串值	<p>用于加密上下文数据的 secret 密钥，以用于性能分析。</p> <p>此字符串值应该具有以下格式：<key1>[,<key2>,...<keyn>]，其中每个键都是随机字符串。通过 REST API 触发性能分析的用户必须在 REST API 调用的标头中设置其中一个键，使其包含此节点的性能分析结果。</p> <p>"enabled"标志和"hmac_keys"配置选项都应设置为启用性能分析。此外，为了跨所有服务生成正确的性能分析信息，OpenStack 项目之间至少需要一个关键。这样可确保它可用于客户端生成 trace，其中包含所有可能的资源的信息。</p>
sentinel_service_name = mymaster	字符串值	<p>Redis sentinel 使用服务名称来识别 master redis 服务。此参数定义名称（例如：sentinal_service_name=mymaster）。</p>
socket_timeout = 0.1	浮点值	<p>Redis sentinel 在连接上提供了一个 timeout 选项。此参数定义该超时（例如：socket_timeout=0.1）。</p>
trace_sqlalchemy = False	布尔值	<p>在服务中启用 SQL 请求性能分析。</p> <p>默认值为 False (SQL 请求不会被跟踪)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用 SQL 请求性能分析。每个 SQL 查询都将是 trace 的一部分，可以通过花费的时间来分析。 ● false：禁用 SQL 请求性能分析。花费的时间仅显示在更高级别的操作上。无法以这种方式分析单个 SQL 查询。

2.1.26. sample_castellan_source

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[sample_castellan_source]` 组下可用的选项。

表 2.25. sample_castellan_source

配置选项 = 默认值	类型	Description
<code>config_file = None</code>	字符串值	到 castellan 配置文件的路径。
<code>driver = None</code>	字符串值	可以加载此配置源的驱动程序名称。
<code>mapping_file = None</code>	字符串值	configuration/castellan_id 映射文件的路径。

2.1.27. sample_remote_file_source

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[sample_remote_file_source]` 组下可用的选项。

表 2.26. sample_remote_file_source

配置选项 = 默认值	类型	Description
<code>ca_path = None</code>	字符串值	CA_BUNDLE 文件或目录的路径，带有可信 CA 的证书。
<code>client_cert = None</code>	字符串值	客户端证书，作为仅包含证书或私钥和证书的单一文件路径。
<code>client_key = None</code>	字符串值	客户端私钥，如果指定了 <code>client_cert</code> ，但没有包含私钥。
<code>driver = None</code>	字符串值	可以加载此配置源的驱动程序名称。
<code>uri = None</code>	URI 值	带有额外配置文件位置的 URI 所需的选项。

2.1.28. service_user

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[service_user]` 组下可用的选项。

表 2.27. service_user

配置选项 = 默认值	类型	Description
<code>auth-url = None</code>	字符串值	身份验证 URL
<code>cafile = None</code>	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
<code>certFile = None</code>	字符串值	PEM 编码的客户端证书证书文件
<code>collect-timing = False</code>	布尔值	收集每个 API 调用计时信息。

配置选项 = 默认值	类型	Description
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
send_service_user_token = False	布尔值	为 True 时，如果向 REST API 发送请求，也会发送服务帐户令牌。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
system-scope = None	字符串值	系统操作的范围
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名

2.1.29. ssl

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[ssl]` 组下可用的选项。

表 2.28. ssl

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	用于验证连接客户端的 CA 证书文件。
cert_file = None	字符串值	安全启动服务器时要使用的证书文件。
密码 = None	字符串值	设置可用密码列表。值应该是 OpenSSL 密码列表格式的字符串。
key_file = None	字符串值	安全启动服务器时使用的私钥文件。
version = None	字符串值	要使用的 SSL 版本（仅在启用了 SSL 时评估）。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1_1 和 TLSv1_2 可能在某些发行版中提供。

2.1.30. vault

下表概述了 `/etc/cinder/cinder.conf` 文件中的 `[vault]` 组下可用的选项。

表 2.29. vault

配置选项 = 默认值	类型	Description
approle_role_id = None	字符串值	Approle role_id 用于使用 vault 进行身份验证
approle_secret_id = None	字符串值	Approle secret_id 用于使用 vault 进行身份验证
kv_mountpoint = secret	字符串值	要使用的 Vault 中的 KV 存储挂载点，例如：secret
kv_version = 2	整数值	要使用的 Vault 中的 KV 版本，例如：2
root_token_id = None	字符串值	vault 的 root 令牌
ssl_ca_cert_file = None	字符串值	ca 证书文件的绝对路径
use_ssl = False	布尔值	SSL Enabled/Disabled
vault_url = http://127.0.0.1:8200	字符串值	使用此端点连接到 Vault，例如："http://127.0.0.1:8200"

第 3 章 GLANCE

下面的章节包含有关 **glance** 服务中的配置选项的信息。

3.1. GLANCE-API.CONF

本节包含 `/etc/glance/glance-api.conf` 文件的选项。

3.1.1. DEFAULT

下表概述了 `/etc/glance/glance-api.conf` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
allow_additional_image_properties = True	布尔值	<p>允许用户为镜像添加额外的/自定义属性。</p> <p>Glance 定义在每个镜像上显示的标准属性集合（在其架构中）。这些属性也称为基本属性。除了这些属性，Glance 允许用户向镜像添加自定义属性。它们被称为附加属性。</p> <p>默认情况下，此配置选项被设置为 True，允许用户添加额外的属性。可以通过 image_property_quota 配置选项控制可添加到镜像中的额外属性数量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • image_property_quota <p>弃用自：U ssuri</p> <p>reason: 这个选项是多余的。通过 <code>image_property_quota</code> 配置选项控制自定义镜像属性的使用。这个选项计划在 Victoria 开发周期中删除。</p>

配置选项 = 默认值	类型	Description
allow_anonymous_access = False	布尔值	<p>允许对未经身份验证的用户进行有限访问。</p> <p>分配一个布尔值来确定取消用户的 API 访问。当设置为 False 时，未经身份验证的用户可以访问 API。当设置为 True 时，未经身份验证的用户可以访问具有只读权限的 API。但是，只有在使用 ContextMiddleware 时才适用。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● None
api_limit_max = 1000	整数值	<p>请求可以返回的最大结果数。</p> <p>如 limit_param_default 的帮助文本中所述，一些请求可能会返回多个结果。要返回的结果数量取决于请求中的 limit 参数或 limit_param_default 配置选项。两个情况下的值都不能大于该配置选项定义的绝对最大值。任何大于这个值的都会修剪到此处定义的最大值。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>把它设置为非常大的值可能会减慢数据库查询并增加响应时间。把它设置为非常低的值可能会导致用户体验不佳。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● limit_param_default

配置选项 = 默认值	类型	Description
backlog = 4096	整数值	<p>设置传入连接请求的数量。</p> <p>提供正整数值来限制 backlog 队列中请求数。默认队列大小为 4096。</p> <p>在与服务器建立连接前，到 TCP 侦听器套接字的传入连接已排队。为 TCP 套接字设置 backlog 可确保传入流量的有限队列大小。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
bind_host = 0.0.0.0	主机地址值	<p>将 glance 服务器绑定到的 IP 地址。</p> <p>提供要将 glance 服务器绑定到的 IP 地址。默认值为 0.0.0.0。</p> <p>编辑这个选项，使服务器能够侦听网卡上的一个特定的 IP 地址。这有助于为服务器选择特定的网络接口。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的 IPv4 地址 ● 有效的 IPv6 地址 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
bind_port = None	端口值	<p>服务器将侦听的端口号。</p> <p>提供将服务器套接字绑定到的有效端口号。然后，将这个端口设置为识别进程并转发到达服务器的网络信息。API 服务器的默认 bind_port 值为 9292，并且 registry 服务器为 9191。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效端口号(0 到 65535) <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
client_socket_timeout = 900	整数值	<p>客户端连接的套接字操作的超时。</p> <p>提供代表时间（以秒为单位）的有效整数值，以设置进入连接关闭前等待的期间。默认值为 900 秒。</p> <p>值零表示等待时间。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 零 ● 正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
control_exchange = openstack	字符串值	默认交换，哪些主题在哪些方面有范围。可以被 transport_url 选项指定的交换名称覆盖。
debug = False	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。

配置选项 = 默认值	类型	Description
<pre>default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpidd=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']</pre>	列表值	<p>以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 log_config_append，则忽略此选项。</p>
<pre>default_publisher_id = image.localhost</pre>	字符串值	<p>用于传出 Glance 通知的默认 publisher_id。</p> <p>这是通知驱动程序将用来识别源自 Glance 服务的事件的消息。通常，这是生成消息的实例的主机名。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何合理的实例标识符，例如：image.host1 <p>相关选项：</p> <ul style="list-style-type: none"> None

配置选项 = 默认值	类型	Description
delayed_delete = False	布尔值	<p>打开/关闭延迟删除。</p> <p>通常，当镜像被删除时，glance-api 服务会将镜像置于 已删除 状态，并同时删除其数据。延迟删除是 Glance 中的一个功能，它可以延迟删除镜像数据直到稍后的时间点（由配置选项 scrub_time 决定）。当打开延迟删除时，glance-api 服务会将镜像置于 pending_delete 状态，并将镜像数据保留在存储后端中，以便镜像清理网络在以后被删除。镜像清理网络将在成功删除镜像数据时将镜像移到 已删除 的状态。</p> <div data-bbox="815 651 922 815" style="float: left; margin-right: 10px;">  </div> <p>注意</p> <p>当打开延迟删除时，镜像清理网络必须作为定期任务运行，以防止后端存储使用不必要的使用量填满。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● scrub_time ● wakeup_time ● scrub_pool_size

配置选项 = 默认值	类型	Description
<p>digest_algorithm = sha256</p>	<p>字符串值</p>	<p>用于数字签名的摘要算法。</p> <p>提供代表用于生成数字签名的摘要算法的字符串值。默认情况下使用 sha256。</p> <p>要获得您的平台上 OpenSSL 版本支持的可用算法列表，请运行以下命令：openssl list-message-digest-algorithms。示例是 <i>sha1</i>、<i>sha256</i> 和 <i>sha512</i>。</p> <div data-bbox="815 577 922 770" style="display: inline-block; vertical-align: top;">  </div> <p>注意</p> <p>digest_algorithm 与 Glance 的镜像签名和验证无关。它仅用于签署通用唯一标识符(UUID)，作为证书文件和密钥文件验证的一部分。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● OpenSSL 消息摘要算法标识符 <p>关系选项：</p> <ul style="list-style-type: none"> ● None


配置选项 = 默认值	类型	Description
disabled_notifications = []	列表值	<p>要禁用的通知列表。</p> <p>指定不应发出的通知列表。可将通知指定为禁用单个事件通知的通知类型，也可以作为通知组前缀来禁用组中的所有事件通知。</p> <p>可能的值：要禁用的单个通知类型或通知组的逗号分隔列表。目前支持的组：</p> <ul style="list-style-type: none"> ● image ● image.member ● task ● metadef_namespace ● metadef_object ● metadef_property ● metadef_resource_type ● metadef_tag <p>For a complete listing and description of each event refer to: http://docs.openstack.org/developer/glance/notifications.html</p> <p>The values must be specified as: <group_name>.<event_name> For example: image.create,task.success,metadef_tag</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● None
enabled_backends = None	dict 值	Key:Value 对存储标识符和存储类型。如果有多个后端，应使用逗号分隔。
enabled_import_methods = ['glance-direct', 'web-download', 'copy-image']	列表值	<p>启用的镜像导入方法列表</p> <p>'glance-direct', 'copy-image' and 'web-download' are enabled by default.</p> <p>Related options: ** [DEFAULT]/node_staging_uri</p>

配置选项 = 默认值	类型	Description
enforce_secure_rbac = False	布尔值	<p>根据 OpenStack 中使用的常见用户角色定义，实施 API 访问。启用此选项规范化特定于项目的读/写操作，如创建私有镜像或更新 member 角色后共享镜像的状态。它还规范了一个可用于特定于项目的 API 操作的只读变体，如在 reader 角色后列出项目中的私有镜像。</p> <p>Operator 应该有机会了解 glance 的新镜像策略，在其部署中审核分配，并使用 keystone 中的默认角色（如 admin、member 和 reader）更新权限。</p> <p>相关选项：</p> <ul style="list-style-type: none"> • [oslo_policy]/enforce_new_defaults <p>弃用自：Wallaby</p> <p>原因：已经引进了此选项，要求操作员根据常见的 RBAC 用户角色选择强制授权，这是 Wallaby 发行版本的 EXPERIMENTAL。此行为将是未来的版本中的默认设置和 STABLE，允许删除这个选项。</p>
executor_thread_pool_size = 64	整数值	当 executor 为 threading 或 eventlet 时，executor 线程池的大小。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。


配置选项 = 默认值	类型	Description
<p>hashing_algorithm = sha512</p>	<p>字符串值</p>	<p>用于计算 <code>os_hash_value</code> 属性的安全哈希算法。</p> <p>此选项配置 Glance "multihash"，它由两个镜像属性组成：<code>os_hash_algo</code> 和 <code>os_hash_value</code>。<code>os_hash_algo</code> 将由此配置选项的值填充，当算法应用到上传或导入的镜像数据时，<code>os_hash_value</code> 将由 <code>hexdigest</code> 填充。</p> <p>该值必须是 <code>python hashlib</code> 库识别的有效安全哈希算法名称。您可以通过检查 Glance 安装中使用的库的 <code>hashlib.algorithms_available</code> 数据成员来确定这些内容。然而，为了实现互操作性，我们建议您使用由 <code>hashlib.algorithms_guaranteed</code> 数据成员提供的安全散列名称集，因为所有平台上的 <code>hashlib</code> 库都可以保证支持这些算法。因此，在本地使用 <code>hashlib</code> 的任何镜像消费者都应能够验证镜像的 <code>os_hash_value</code>。</p> <p><code>sha512</code> 的默认值是一个性能安全哈希算法。</p> <p>如果此选项配置不正确，任何尝试存储镜像数据都将失败。因此，我们建议使用默认值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● <code>Python hashlib</code> 库识别的任何安全哈希算法名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>None</code>
<p>http_keepalive = True</p>	<p>布尔值</p>	<p>为 HTTP over TCP 设置 keep alive 选项。</p> <p>提供布尔值，以确定保留数据包的发送。如果设置为 False，服务器会返回标头 "Connection: close"。如果设置为 True，服务器会在响应中返回一个 "Connection: Keep-Alive"。这可为 HTTP 对话保留相同的 TCP 连接，而不是为每个新请求打开一个新的连接。</p> <p>如果在收到响应并且客户端成功读取后需要显式关闭客户端套接字连接，则必须将此选项设置为 False。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● <code>True</code> ● <code>False</code> <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>None</code>

配置选项 = 默认值	类型	Description
image_cache_dir = None	字符串值	<p>镜像缓存的基础目录。</p> <p>这是缓存镜像数据的位置，并从其中提供。所有缓存的镜像都直接存储在这个目录中。此目录还包含三个子目录，即 不完整、无效 和 队列。</p> <p>incomplete 子目录是下载镜像的暂存区域。镜像首次下载到此目录。当镜像下载成功时，它会移到基础目录。但是，如果下载失败，部分下载的镜像文件将移到 无效的 子目录中。</p> <p>queue 子目录用于排队镜像以进行下载。这主要由 cache-prefetcher 使用，它可以作为定期任务调度，如 cache-pruner 和 cache-cleaner，以在使用前缓存镜像。在收到缓存镜像的请求后，Glance 将把文件放置在 'queue 目录中，其镜像 ID 为文件名。cache-prefetcher 在运行时，轮询 队列 目录中的文件，并按照创建的顺序开始下载它们。下载成功后，从 队列 目录中删除零化文件。如果下载失败，则零化文件会保留，并将在下次运行 cache-prefetcher 时重试。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● image_cache_sqlite_db

配置选项 = 默认值	类型	Description
<p>image_cache_driver = sqlite</p>	<p>字符串值</p>	<p>用于镜像缓存管理的驱动程序。</p> <p>此配置选项提供在不同镜像缓存驱动程序之间进行选择的灵活性。image-cache 驱动程序负责提供 image-cache 的基本功能，如从缓存写入/读取镜像，跟踪缓存镜像的年龄和使用，提供缓存镜像列表、缓存获取大小、用于缓存和清理缓存等。</p> <p>驱动程序的基本功能在基本类 glance.image_cache.drivers.base.Driver 中定义。所有镜像缓存驱动程序（现有驱动程序和潜在版本）都必须实现这个接口。目前可用的驱动程序有 sqlite 和 xattr。这些驱动程序主要有所不同，它们存储缓存镜像的信息：</p> <ul style="list-style-type: none"> ● sqlite 驱动程序使用 sqlite 数据库（位于本地每个 glance 节点上）来跟踪缓存镜像的使用。 ● xattr 驱动程序使用文件的扩展属性来存储此信息。它还需要一个在访问时对文件设置 atime 的文件系统。 <p>可能的值：</p> <ul style="list-style-type: none"> ● SQLite ● xattr <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
image_cache_max_size = 10737418240	整数值	<p>缓存大小上限（以字节为单位），这会清理镜像缓存。</p>  <p>注意</p> <p>这只是 cache-pruner 操作的阈值。它不是镜像缓存永远不会增长的硬限制。实际上，根据 cache-pruner 的运行频率以及缓存填满的速度，镜像缓存不能超过此处指定的大小。因此，必须小心才能适当地调度 cache-pruner 并设定这个限制。</p> <p>Glance 在下载镜像时缓存镜像。因此，随着下载数量增加，镜像缓存的大小会随时间而增长。要将缓存大小保持不管理，建议将 cache-pruner 作为定期任务运行。当缓存修剪器启动时，它会比较镜像缓存的当前大小，并在镜像缓存超过此处指定的大小时触发清理。清理后，缓存的大小应小于或等于此处指定的大小。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何非负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
image_cache_sqlite_db = cache.db	字符串值	<p>用于镜像缓存管理的 sqlite 文件数据库的相对路径。</p> <p>这是到 sqlite 文件数据库的相对路径，用于跟踪镜像缓存的年龄和使用统计信息。该路径相对于镜像缓存基础目录，由配置选项 image_cache_dir 指定。</p> <p>这是一个轻量级数据库，只有一个表。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 到 sqlite 文件数据库的有效相对路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● image_cache_dir

配置选项 = 默认值	类型	Description
image_cache_stall_time = 86400	整数值	<p>不完整的镜像保留在缓存中的时间（以秒为单位）。</p> <p>不完整的镜像是下载正在进行的镜像。详情请查看配置选项 image_cache_dir 描述。有时，由于各种原因，下载可能会挂起，未下载的镜像保留在 不完整的 目录中。此配置选项设置在清理前镜像应保留在 不完整 目录中的时间限制。当不完整的镜像花费的时间超过此处指定的时间后，将在下次运行时通过 <code>cache-cleaner</code> 删除。</p> <p>建议将 <code>cache-cleaner</code> 作为 Glance API 节点上的定期任务运行，以保持磁盘空间中不完整的镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何非负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
image_location_quota = 10	整数值	<p>镜像上允许的最大位置数。</p> <p>任何负值都解释为无限。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● None
image_member_quota = 128	整数值	<p>每个镜像的最大镜像成员数量。</p> <p>这限制了镜像可以被共享的最大用户。任何负值都解释为无限。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
image_property_quota = 128	整数值	<p>镜像上允许的最大属性数。</p> <p>这对镜像可以具有的额外属性数量强制实施上限。任何负值都解释为无限。</p> <div style="display: flex; align-items: flex-start;">  <div> <p>注意</p> <p>如果禁用附加属性，则这不会影响。请参阅 allow_additional_image_properties。</p> </div> </div> <p>相关选项：</p> <ul style="list-style-type: none"> ● allow_additional_image_properties
image_size_cap = 1099511627776	整数值	<p>用户可以上传的最大镜像大小（以字节为单位）。</p> <p>大于此处提到的大小的镜像上传会导致镜像创建失败。此配置选项默认为 1099511627776 字节(1 TiB)。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 这个值只有在仔细考虑后才会增加，且必须设置小于或等于 8 EiB (9223372036854775808)。 ● 必须谨慎考虑后端存储容量设置这个值。把它设置为非常低的值可能会导致大量镜像失败。然后，将其设置为一个非常大的值可能会导致存储更快地消耗。因此，这必须根据创建的镜像的性质和可用的存储容量来设置。 <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何小于或等于 9223372036854775808 的正数
image_tag_quota = 128	整数值	<p>镜像上允许的最大标签数。</p> <p>任何负值都解释为无限。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● None
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	<p>通过日志消息传递的实例格式。</p>

配置选项 = 默认值	类型	Description
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。
limit_param_default = 25	整数值	<p>为请求返回的默认结果数。</p> <p>对特定 API 请求的响应（如列出镜像）可能会返回多个项目。通过在 API 请求中指定 limit 参数，可以显式控制返回的结果数量。但是，如果没有指定 limit 参数，则此配置值将用作任何 API 请求返回的默认结果数。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 此配置选项的值可能不大于 api_limit_max 指定的值。 ● 把它设置为非常大的值可能会减慢数据库查询并增加响应时间。把它设置为非常低的值可能会导致用户体验不佳。 <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>api_limit_max</code>
location_strategy = location_order	字符串值	<p>决定镜像位置首选项顺序的策略。</p> <p>此配置选项指示决定镜像位置必须被访问才能提供镜像数据的策略。Glance 随后从找到的第一个响应活动位置检索镜像数据。</p> <p>此选项取两个可能的值 location_order 和 store_type 中的一个。默认值为 location_order，这表示镜像数据按照存储在 Glance 中的顺序使用位置来提供。store_type 值根据存储后端被列为配置选项 store_type_preference 的以逗号分隔的列表的顺序来设置镜像位置首选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● <code>location_order</code> ● <code>store_type</code> <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>store_type_preference</code>

配置选项 = 默认值	类型	Description
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 log-date-format）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 DEBUG 时，要附加到日志消息的其他数据。 oslo_log.formatters.ContextFormatter 使用
logging_default_format_s tring = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用

配置选项 = 默认值	类型	Description
logging_exception_prefix = <code>%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s</code>	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_format = <code>%(user) s %(tenant) s %(domain) s %(user_domain) s %(project_domain) s</code>	字符串值	定义 logging_context_format_string 中使用的 <code>%(user_identity) s</code> 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_header_line = 16384	整数值	<p>消息标头的最大行大小。</p> <p>提供一个代表长度的整数值来限制消息标头的大小。默认值为 16384。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>在使用大型令牌时（通常是由 Keystone v3 API 和大型服务目录生成的），可能需要增加 max_header_line。但是，请记住 max_header_line 的较大的值会填充日志。</p> </div> </div> <p>将 max_header_line 设置为 0，为消息标头的行大小设置 no limit。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0 ● 正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。

配置选项 = 默认值	类型	Description
max_request_id_length = 64	整数值	<p>限制请求 ID 长度。</p> <p>提供一个整数值，将请求 ID 的长度限制为指定的长度。默认值为 64。用户可以将其改为 0 到 16384 之间的任意 ineteger 值，但请注意，较大的值可能会填充日志。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0 到 16384 之间的整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
metadata_encryption_key = None	字符串值	<p>用于加密存储位置元数据的 AES 密钥。</p> <p>提供代表用于加密 Glance 存储元数据的 AES 密码的字符串值。</p> <div style="display: flex; align-items: flex-start;">  <div> <p>注意</p> <p>要使用的 AES 密钥必须设置为随机字符串，长度为 16、24 或 32 字节。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表有效 AES 密钥的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
<p>node_staging_uri = file:///tmp/staging/</p>	字符串值	<p>URL 提供临时存储临时数据的位置</p> <p>此选项仅用于 Glance 内部用途。Glance 将在镜像导入过程中保存用户上传的镜像数据到 <i>暂存</i> 端点。</p> <p>这个选项不会以任何方式更改 <i>staging</i> API 端点。</p> <p> 注意</p> <p>不建议使用与 [task]/work_dir 相同的路径</p> <p> 注意</p> <p><i>file://<absolute-directory-path ></i> 是现在唯一支持的 <i>api_image_import</i> 流。</p> <p> 注意</p> <p>暂存路径必须位于所有 Glance API 节点可用的共享文件系统上。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 以 <i>file://</i> 开始的字符串，后跟绝对 FS 路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● [task]/work_dir

配置选项 = 默认值	类型	Description
property_protection_file = None	字符串值	<p>属性保护文件的位置。</p> <p>提供到属性保护文件的有效路径，其中包含属性保护的规则以及与其关联的 roles/policies。</p> <p>属性保护文件在设置时，限制要创建、读取、更新和/或删除由角色或策略标识的特定用户集的 Glance 镜像属性。如果没有设置此配置选项，默认情况下不会强制执行属性保护。如果指定了值且未找到文件，则 glance-api 服务将无法启动。有关属性保护的更多信息，请参阅：</p> <p>https://docs.openstack.org/glance/latest/admin/property-protections.html</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 空字符串 ● 属性保护配置文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● property_protection_rule_format
property_protection_rule_format = roles	字符串值	<p>属性保护的规则格式。</p> <p>提供所需的方法来设置 Glance 镜像属性的保护。两个允许的值是 角色和策略。默认值为 roles。</p> <p>如果值为 roles，则属性保护文件必须包含以逗号分隔的用户角色列表，指示每个受保护的属性上的 CRUD 操作的权限。如果设置为 policies，则使用 policy.yaml 中定义的策略来表达每个 CRUD 操作的属性保护。有关如何根据 角色或策略 强制实施属性保护的示例，请访问：</p> <p>https://docs.openstack.org/glance/latest/admin/property-protections.html#examples</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● roles ● policies <p>相关选项：</p> <ul style="list-style-type: none"> ● property_protection_file

配置选项 = 默认值	类型	Description
public_endpoint = None	字符串值	<p>用于 Glance 版本响应的公共 url 端点。</p> <p>这是 Glance "versions" 响应中出现的公共 url 端点。如果没有指定值，则版本响应中显示的端点是运行 API 服务的主机。如果 API 服务在代理后面运行，请将端点更改为代表代理 URL。如果服务在负载均衡器后面运行，请为这个值添加负载均衡器的 URL。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • None • 代理 URL • 负载均衡器 URL <p>相关选项：</p> <ul style="list-style-type: none"> • None
publish_errors = False	布尔值	<p>启用或禁用错误事件的发布。</p>
pydev_worker_debug_host = None	主机地址值	<p>pydev 服务器的主机地址。</p> <p>提供代表用于调试的 pydev 服务器的主机名或 IP 的字符串值。pydev 服务器侦听此地址上的调试连接，促进 Glance 中的远程调试。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 有效主机名 • 有效的 IP 地址 <p>相关选项：</p> <ul style="list-style-type: none"> • None
pydev_worker_debug_port = 5678	端口值	<p>pydev 服务器将侦听的端口号。</p> <p>提供将 pydev 服务器绑定到的端口号。pydev 进程接受此端口上的调试连接，并有助于在 Glance 中进行远程调试。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 有效端口号 <p>相关选项：</p> <ul style="list-style-type: none"> • None

配置选项 = 默认值	类型	Description
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
rpc_conn_pool_size = 30	整数值	RPC 连接池的大小。
rpc_ping_enabled = False	布尔值	添加端点以回答 ping 调用。端点名为 oslo_rpc_server_ping
rpc_response_timeout = 60	整数值	从调用等待响应的秒数。
scrub_pool_size = 1	整数值	<p>用于清理镜像的线程池的大小。</p> <p>当大量镜像可用于清理时，可以并行清理镜像，以便清理队列保持控制，后端存储会及时回收。此配置选项表示并行清理的最大镜像数。默认值为 one，表示串行清理。上面的任何值都表示并行清理。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何非零正整数 <p>相关选项：</p> <ul style="list-style-type: none"> delayed_delete

配置选项 = 默认值	类型	Description
scrub_time = 0	整数值	<p>延迟镜像清理的时间（以秒为单位）。</p> <p>当延迟删除被打开时，镜像会在删除时放入 pending_delete 状态，直到清理时间删除其镜像数据。通常，在镜像进入 pending_delete 状态后，就可以进行清理。但是，清理可能会延迟，直到稍后使用这个配置选项为止。此选项表示镜像在可用于清理前处于 pending_delete 状态的时间周期。</p> <p>务必要意识到，这对存储有影响。scrub_time 越大，从已删除镜像回收后端存储所需的时间越长。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何非负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● delayed_delete
secure_proxy_ssl_header = None	字符串值	<p>用于确定原始请求的方案的 HTTP 标头，即使它是由 SSL 终止代理删除的。典型的值为 "HTTP_X_FORWARDED_PROTO"。</p>

配置选项 = 默认值	类型	Description
<p>show_image_direct_url = False</p>	<p>布尔值</p>	<p>在返回镜像时显示直接镜像位置。</p> <p>此配置选项指示在向用户返回镜像详情时是否显示直接镜像位置。直接镜像位置是镜像数据存储在后端存储中的位置。此镜像位置显示在镜像属性 direct_url 下。</p> <p>当镜像存在多个镜像位置时，会根据配置选项 location_strategy 指示的位置策略显示最佳位置。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 显示镜像位置可能会显示 GRAVE SECURITY RISK，因为镜像位置有时可以包含凭证。因此，默认设置为 False。使用 EXTREME CAUTION 把它设置为 True，仅 IF 您知道自己正在做什么！ ● 如果运算符希望避免向用户显示任何镜像位置，则此选项和 show_multiple_locations MUST 都设为 False。 <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● show_multiple_locations ● location_strategy

配置选项 = 默认值	类型	Description
show_multiple_locations = False	布尔值	<p>显示返回镜像时的所有镜像位置。</p> <p>此配置选项指示在向用户返回镜像详情时是否显示所有镜像位置。当镜像存在多个镜像位置时，根据配置 opt location_strategy 指示的位置策略进行排序。镜像位置显示在镜像属性 位置 下。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 显示镜像位置可能会显示 GRAVE SECURITY RISK，因为镜像位置有时可以包含凭证。因此，默认设置为 False。使用 EXTREME CAUTION 把它设置为 True，仅 IF 您知道自己正在做什么！ ● 如需更多信息，请参阅 https://wiki.openstack.org/wiki/OSSN/OSSN-0065。 ● 如果运算符希望避免向用户显示任何镜像位置，则此选项和 show_image_direct_url MUST 都被设置为 False。 <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● show_image_direct_url ● location_strategy <p>弃用自：Newton</p> <p>*reason:*Use of option（自 Newton 起已弃用）是一个安全风险，当我们找出满足目前需要它的用例后，将删除它。之前宣布可使用策略以更高的粒度实现相同的功能不正确。您无法在当前时间通过策略配置临时解决此选项，但这是我们认为该修复将采取的方向。请留意 Glance 发行注记，以保持在这个问题的最新进度。</p>
syslog-log-facility = LOG_USER	字符串值	<p>syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。</p>

配置选项 = 默认值	类型	Description
tcp_keepidle = 600	整数值	<p>在重新检查连接前设置等待时间。</p> <p>提供一个正整数值，代表时间（以秒为单位）在 TCP 保留实时数据包可以发送到主机前作为空闲等待时间设置。默认值为 600 秒。</p> <p>设置 tcp_keepidle 有助于定期验证连接是否完好，并防止频繁建立 TCP 连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 正整数值，代表时间（以秒为单位） <p>相关选项：</p> <ul style="list-style-type: none"> None
transport_url = rabbit://	字符串值	<p>用于连接到消息传递后端的网络地址和可选用户凭证，采用 URL 格式。预期格式为：</p> <pre>driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query</pre> <p>示例：rabbit://rabbitmq:password@127.0.0.1:5672//</p> <p>有关 URL 中字段的详情，请查看 oslo_messaging.TransportURL 文档</p>
use-journal = False	布尔值	<p>为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 <code>log_config_append</code>，此选项将被忽略。</p>
use-json = False	布尔值	<p>使用 JSON 格式进行日志记录。如果设置了 <code>log_config_append</code>，则忽略此选项。</p>
use-syslog = False	布尔值	<p>使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 <code>log_config_append</code>，则忽略此选项。</p>
use_eventlog = False	布尔值	<p>日志输出到 Windows 事件日志。</p>
use_stderr = False	布尔值	<p>将输出输出到标准错误。如果设置了 <code>log_config_append</code>，则忽略此选项。</p>

配置选项 = 默认值	类型	Description
user_storage_quota = 0	字符串值	<p>每个租户的最大镜像存储量。</p> <p>这会强制对所有存储中的所有租户镜像使用的累积存储进行上限。这是每个租户的限制。</p> <p>此配置选项的默认单位是字节。但是，可以使用区分大小写的字面 B、KB、MB、GB 和 TB 指定存储单元，分别代表 Bytes、KiloBytes、MegaBytes、GigaBytes 和 TeraBytes。请注意，值和单元之间不应有任何空格。值 0 表示没有配额强制。负值无效，并导致错误。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 一个字符串，它是一个非负整数的有效串联，代表存储值，以及代表上面提到的存储单元的可选字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> None
watch-log-file = False	布尔值	<p>使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 <code>log_file</code> 选项并使用 Linux 平台时，才有意义。如果设置了 <code>log_config_append</code>，则忽略此选项。</p>
worker_self_reference_url = None	字符串值	<p>此 worker 的 URL。</p> <p>如果设置了此设置，其他 glance 工作程序将了解如何直接联系此 worker。对于镜像导入，单个 worker 分阶段镜像和其他 worker 需要能够将导入请求代理到正确的请求。</p> <p>如果未设置，这被视为 public_endpoint，这通常会设置为所有 worker 上的相同值，从而有效地禁用代理行为。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 可以从其他 worker 访问此 worker 的 URL <p>相关选项：</p> <ul style="list-style-type: none"> <code>public_endpoint</code>

配置选项 = 默认值	类型	Description
Worker = None	整数值	<p>要启动的 Glance worker 进程数量。</p> <p>提供非负的整数值，将子进程 worker 的数量设置为服务请求。默认情况下，可用的 CPU 数量被设置为 worker 限制为 8。例如，如果处理器数是 6，则使用 6 个 worker，如果处理器数为 24 个，则将只使用 8 个 worker。如果配置了 24 个 worker，则限制仅应用到默认值（如果配置了 24 个 worker）。</p> <p>每个 worker 进程都进行侦听配置文件中设置的端口，并包含 greenthread 池 1000。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>将 worker 数量设置为零，会触发单个 API 进程创建大小为 1000 的 greenthread 池。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0 ● 正整数值（通常等于 CPU 的数量） <p>相关选项：</p> <ul style="list-style-type: none"> ● None

3.1.2. cinder

下表概述了 `/etc/glance/glance-api.conf` 文件中的 **[cinder]** 组下可用的选项。

表 3.1. cinder

配置选项 = 默认值	类型	Description
cinder_api_insecure = False	布尔值	<p>允许对 cinder 执行不安全的 SSL 请求。</p> <p>如果此选项被设置为 True，则使用 cinder_ca_certificates_file 选项指定的 CA 证书文件验证 HTTPS 端点连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_ca_certificates_file

配置选项 = 默认值	类型	Description
cinder_ca_certificates_file = None	字符串值	<p>用于 Cinder 客户端请求的 CA 证书文件的位置。</p> <p>指定的 CA 证书文件（如果设置）用于通过 HTTPS 端点验证 Cinder 连接。如果端点是 HTTP，则忽略这个值。cinder_api_insecure 必须设置为 True 以启用验证。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● ca 证书文件的路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_api_insecure
cinder_catalog_info = volumev3::publicURL	字符串值	<p>在服务目录中查找 cinder 时匹配的信息。</p> <p>如果没有设置 cinder_endpoint_template，并且任何 cinder_store_auth_address，cinder_store_user_name，cinder_store_project_name，cinder_store_password 没有设置，cinder 存储将使用此信息从当前上下文中的服务目录中查找 Cinder 端点。如果设置，将考虑 cinder_os_region_name 来获取适当的端点。</p> <p>服务目录可以通过 openstack catalog list 命令列出。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 应指定以下形式的字符串：< service_type>:< service_name>:< interface > At least service_type 和 interface。 service_name 可以省略。 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_os_region_name ● cinder_endpoint_template ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password

配置选项 = 默认值	类型	Description
cinder_endpoint_template = None	字符串值	<p>使用 cinder 端点的模板覆盖服务目录查找。</p> <p>设置此选项时，这个值用于生成 cinder 端点，而不是从服务目录查找。如果指定了 cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, 和 cinder_store_password, 则忽略这个值。</p> <p>如果设置了此配置选项，则忽略 cinder_catalog_info。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● cinder 端点的 URL 模板字符串，其中 %%(tenant) 替换为当前的租户（项目）名称。例如： http://cinder.openstack.example.org/v2/%%(tenant)s <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password ● cinder_catalog_info
cinder_enforce_multipath = False	布尔值	<p>如果设置为 True，则当 multipathd 未运行时，镜像传输的卷附加将中止。否则，它将回退到单一路径。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True 或 False <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_use_multipath
cinder_http_retries = 3	整数值	<p>失败 http 调用的 cinderclient 重试次数。</p> <p>当调用被任何错误失败时，cinderclient 会在睡眠几秒钟后重试调用。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
cinder_mount_point_base = /var/lib/glance/mnt	字符串值	<p>NFS 卷挂载在 glance 节点上的目录。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表挂载点的绝对路径的字符串。
cinder_os_region_name = None	字符串值	<p>从服务目录查找 cinder 服务的区域名称。</p> <p>这仅在使用 cinder_catalog_info 来确定端点时使用。如果设置，则此节点的 cinder 端点的查找将过滤到指定的区域。当目录中列出了多个区域时，它很有用。如果没有设置，则会从每个区域查找端点。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 是有效区域名称的字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_catalog_info
cinder_state_transition_timeout = 300	整数值	<p>等待 cinder 卷转换完成的时间（以秒为单位）。</p> <p>当创建、删除或附加到 glance 节点以读/写卷数据时，卷的状态将改变。例如，新创建的卷状态会在 创建 过程完成后变为 available。这将指定等待状态更改的最长时间。如果在等待时发生超时，或者状态更改为非预期的值（如 错误），则镜像创建会失败。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
cinder_store_auth_addresses = None	字符串值	<p>cinder 身份验证服务正在侦听的地址。</p> <p>当指定所有 cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, 和 cinder_store_password 选项时, 指定的值始终用于身份验证。这有助于通过将镜像服务存储在特定于镜像服务的项目/租户中, 从用户隐藏镜像卷。它还允许用户在 glance 的 ACL 控制下与其他项目共享镜像卷。</p> <p>如果没有设置任何一个选项, 则 cinder 端点会从服务目录中查找, 并使用当前上下文的用户和项目。</p> <p>可能的值 :</p> <ul style="list-style-type: none"> 有效的身份验证服务地址, 例如 : http://openstack.example.org/identity/v2.0 <p>相关选项 :</p> <ul style="list-style-type: none"> cinder_store_user_name cinder_store_password cinder_store_project_name
cinder_store_password = None	字符串值	<p>用于针对 Cinder 进行身份验证的用户密码。</p> <p>这必须与以下所有相关选项一起使用。如果未指定其中任何一个, 则使用当前上下文的用户。</p> <p>可能的值 :</p> <ul style="list-style-type: none"> 由 cinder_store_user_name 指定的用户的有效密码 <p>相关选项 :</p> <ul style="list-style-type: none"> cinder_store_auth_address cinder_store_user_name cinder_store_project_name

配置选项 = 默认值	类型	Description
cinder_store_project_name = None	字符串值	<p>镜像卷的项目名称存储在 cinder 中。</p> <p>如果没有设置此配置选项，则使用当前上下文中的项目。</p> <p>这必须与以下所有相关选项一起使用。如果未指定其中任何一个，则使用当前上下文的项目。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效项目名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_password
cinder_store_user_name = None	字符串值	<p>要针对 cinder 进行身份验证的用户名。</p> <p>这必须与以下所有相关选项一起使用。如果未指定其中任何一个，则使用当前上下文的用户。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的用户名 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_password ● cinder_store_project_name
cinder_use_multipath = False	布尔值	<p>在部署中支持或不支持识别 mutipath 的标记。</p> <p>如果不支持多路径，则将其设置为 False。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True 或 False <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_enforce_multipath

配置选项 = 默认值	类型	Description
cinder_volume_type = None	字符串值	<p>在 cinder 中用于创建卷的卷类型。</p> <p>有些 Cinder 后端可以有几个卷类型来优化存储使用。添加此选项可让操作器在 cinder 中选择特定卷类型，该类型可以针对镜像进行了优化。</p> <p>如果没有设置，则 cinder 配置中指定的默认卷类型将用于创建卷。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 来自 cinder 的有效卷类型 <p>相关选项：</p> <ul style="list-style-type: none"> None <div style="display: flex; align-items: flex-start;">  <div> <p>注意</p> <p>您不能使用与 NFS 后端关联的加密 volume_type。当 glance_store 试图写入或访问存储在该卷中的镜像数据时，存储在 NFS 后端上的加密卷将引发异常。请参考 Cinder 管理员确定适当的 volume_type。</p> </div> </div>
rootwrap_config = /etc/glance/rootwrap.conf	字符串值	<p>用于以 root 身份运行命令的 rootwrap 配置文件的路径。</p> <p>cinder 存储需要 root 特权才能运行镜像卷（用于连接 iSCSI/FC 卷并读取/写入卷数据等）。配置文件应允许 cinder 存储和 os-brick 库所需的命令。</p> <p>可能的值：</p> <ul style="list-style-type: none"> rootwrap 配置文件的路径 <p>相关选项：</p> <ul style="list-style-type: none"> None

3.1.3. CORS

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[cors]` 组下可用的选项。

表 3.2. CORS

配置选项 = 默认值	类型	Description
allow_credentials = True	布尔值	指明实际请求可以包含用户凭证

配置选项 = 默认值	类型	Description
allow_headers = ['Content-MD5', 'X-Image-Meta-Checksum', 'X-Storage-Token', 'Accept-Encoding', 'X-Auth-Token', 'X-Identity-Status', 'X-Roles', 'X-Service-Catalog', 'X-User-Id', 'X-Tenant-Id', 'X-OpenStack-Request-ID']	列表值	指定在实际请求过程中可以使用哪些标头字段名称。
allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH']	列表值	指明在实际请求期间可以使用哪些方法。
allowed_origin = None	列表值	指明此资源是否可以与请求"origin"标头中收到的域共享。格式： <code>"<protocol>://<host>[:<port>]"</code> ，没有尾部斜杠。示例： https://horizon.example.com
expose_headers = ['X-Image-Meta-Checksum', 'X-Auth-Token', 'X-Subject-Token', 'X-Service-Token', 'X-OpenStack-Request-ID']	列表值	指明哪些标头可以安全地公开给 API。默认为 HTTP Simple Headers。
max_age = 3600	整数值	CORS preflight 请求的最大缓存期限。

3.1.4. database

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[database]` 组下可用的选项。

表 3.3. database

配置选项 = 默认值	类型	Description
backend = sqlalchemy	字符串值	用于数据库的后端。
connection = None	字符串值	用于连接到数据库的 SQLAlchemy 连接字符串。
connection_debug = 0	整数值	SQL 调试信息的详细程度：0=None, 100=Everything。
<code>connection_parameters =</code>	字符串值	用于连接时附加到连接 URL 的可选 URL 参数；指定 <code>param1=value1&param2=value2&...</code>

配置选项 = 默认值	类型	Description
connection_recycle_time = 3600	整数值	连接池中存在的连接的时间超过这个秒数，下一次从池中签出它们的时间将替换为新的连接。
connection_trace = False	布尔值	将 Python 堆栈跟踪作为注释字符串添加到 SQL。
db_inc_retry_interval = True	布尔值	如果为 True，将数据库操作的重试间隔增加到 db_max_retry_interval。
db_max_retries = 20	整数值	在引发错误前出现连接错误或死锁错误时的最大重试次数。设置为 -1 以指定无限重试计数。
db_max_retry_interval = 10	整数值	如果设置了 db_inc_retry_interval，则数据库操作重试之间的最大秒数。
db_retry_interval = 1	整数值	数据库事务重试之间的秒数。
max_overflow = 50	整数值	如果设置，请将这个值用于 SQLAlchemy 的 max_overflow。
max_pool_size = 5	整数值	在池中保留的最大 SQL 连接数。将值设为 0 表示没有限制。
max_retries = 10	整数值	启动期间数据库连接重试的最大数量。设置为 -1 以指定无限重试计数。
mysql_enable_ndb = False	布尔值	如果为 True，则透明地启用对处理 MySQL 集群 (NDB) 的支持。
mysql_sql_mode = TRADITIONAL	字符串值	用于 MySQL 会话的 SQL 模式。这个选项（包括默认选项）会覆盖任何 server-set SQL 模式。要使用服务器配置设置的任何 SQL 模式，请将其设置为 no 值。 示例：mysql_sql_mode=
pool_timeout = None	整数值	如果设置，请将这个值用于 SQLAlchemy 的 pool_timeout。
retry_interval = 10	整数值	重新尝试打开 SQL 连接之间的间隔。
slave_connection = None	字符串值	用于连接到从数据库的 SQLAlchemy 连接字符串。
sqlite_synchronous = True	布尔值	如果为 True，则 SQLite 使用同步模式。
use_db_reconnect = False	布尔值	启用在连接丢失时进行数据库重新连接的实验性使用。

配置选项 = 默认值	类型	Description
use_tpool = False	布尔值	启用所有 DB API 调用的线程池的实验性使用

3.1.5. file

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[file]` 组下可用的选项。

表 3.4. file

配置选项 = 默认值	类型	Description
filesystem_store_chunk_size = 65536	整数值	<p>块大小，以字节为单位。</p> <p>读取或写入镜像文件时使用的块大小。增加这个值可以提高吞吐量，但在处理大量请求时，它也可能稍微增加内存用量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
<p>filesystem_store_datadir = /var/lib/glance/images</p>	<p>字符串值</p>	<p>文件系统后端存储写入镜像的目录。</p> <p>启动时，Glance 会创建该目录（如果不存在），并验证在其下运行 glance-api 的用户的写入访问权限。如果写入访问不可用，则引发 BadStoreConfiguration 异常，并且文件系统存储可能无法添加新镜像。</p> <div data-bbox="815 510 922 824" style="border: 1px solid gray; padding: 5px; width: fit-content;">  </div> <p>注意</p> <p>此目录仅在将文件系统存储用作存储后端时使用。filesystem_store_datadir 或 filesystem_store_datadirs 选项必须在 glance-api.conf 中指定。如果指定了这两个选项，则会引发 BadStoreConfiguration，并且文件系统存储可能无法添加新镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 到目录的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● filesystem_store_datadirs ● filesystem_store_file_perm

配置选项 = 默认值	类型	Description
filesystem_store_datadirs = None	多值	<p>文件系统后端存储写入镜像的目录及其优先级列表。</p> <p>文件系统存储可以配置为将镜像存储在多个目录中，而不是使用由 filesystem_store_datadir 配置选项指定的单个目录。使用多个目录时，可以为每个目录赋予一个可选优先级，以指定应使用它们的首选项顺序。priority 是一个整数，它将与目录路径连接，冒号越高，代表优先级更高。当两个目录具有相同的优先级时，会使用具有最多可用空间的目录。如果没有指定优先级，则默认为零。</p> <p>有关使用多个存储目录配置文件系统存储的更多信息，请参阅 https://docs.openstack.org/glance/latest/configuration/configuring.html</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>此目录仅在将文件系统存储用作存储后端时使用。filesystem_store_datadir 或 filesystem_store_datadirs 选项必须在 glance-api.conf 中指定。如果指定了这两个选项，则会引发 BadStoreConfiguration，并且文件系统存储可能无法添加新镜像。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 以下格式的字符串列表： <ul style="list-style-type: none"> ○ <a valid directory path>:<optional integer priority> <p>相关选项：</p> <ul style="list-style-type: none"> ● filesystem_store_datadir ● filesystem_store_file_perm

配置选项 = 默认值	类型	Description
filesystem_store_file_permissions = 0	整数值	<p>镜像文件的文件访问权限。</p> <p>为镜像数据设置预期的文件访问权限。这提供了一种方式，可以让其他服务（如 Nova）直接使用文件系统存储中的镜像。运行要被授予访问权限的服务的用户，可以成为拥有创建文件的组的成员。为这个配置选项分配一个小于或等于 0 的值表示没有对默认权限进行任何更改。这个值将被解码为八进制数。</p> <p>如需更多信息，请参阅文档 https://docs.openstack.org/glance/latest/configuration/configuring.html</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的文件访问权限 ● 零 ● 任何负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
filesystem_store_metadata_file = None	字符串值	<p>文件系统存储元数据文件。</p> <p>包含与文件系统存储关联的任何位置返回元数据的文件路径。设置此选项后，它仅用于之后创建的新镜像 - 以前现有镜像不受影响。</p> <p>该文件必须包含有效的 JSON 对象。该对象应包含键 id 和 mountpoint。两个键的值都应该是一个字符串。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 存储元数据文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
filesystem_thin_provisioning = False	布尔值	<p>在这个后端中启用或不精简置备。</p> <p>这个配置选项启用了在文件系统中没有真正写入 null 字节序列的功能，文件系统会自动解释为 null 字节，且不会实际消耗您的存储。启用此功能也会加快镜像上传速度，并节省后端中的空间，因为不会通过网络发送 null 字节序列。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • None

3.1.6. glance.store.http.store

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[glance.store.http.store]` 组下可用的选项。

表 3.5. glance.store.http.store

配置选项 = 默认值	类型	Description
http_proxy_information = {}	dict 值	<p>用于连接到远程服务器的 http/https 代理信息。</p> <p>这个配置选项指定应该用来连接到远程服务器的 http/https 代理信息。代理信息应该是方案和代理的键值对，如 <code>http:10.0.0.1:3128</code>。您还可以通过使用逗号分隔键值对（如 <code>http:10.0.0.1:3128, https:10.0.0.1:1080</code>）来指定多个方案的代理。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 如前文所述，以逗号分隔的 <code>scheme:proxy</code> 对列表 <p>相关选项：</p> <ul style="list-style-type: none"> • None

配置选项 = 默认值	类型	Description
https_ca_certificates_file = None	字符串值	<p>CA 捆绑包文件的路径。</p> <p>此配置选项可让操作员使用自定义证书颁发机构文件来验证远程服务器证书。如果设置了这个选项，则忽略 https_insecure 选项，指定的 CA 文件将用于验证服务器证书并建立到服务器的安全连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● CA 文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● https_insecure
https_insecure = True	布尔值	<p>设置远程服务器证书的验证。</p> <p>此配置选项使用布尔值来确定是否验证远程服务器证书。如果设置为 True，则不会验证远程服务器证书。如果选项设为 False，则使用默认 CA 信任存储进行验证。</p> <p>如果设置了 https_ca_certificates_file，则忽略这个选项。然后，使用 https_ca_certificates_file 选项指定的文件验证远程服务器证书。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● https_ca_certificates_file

3.1.7. glance.store.rbd.store

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[glance.store.rbd.store]` 组下可用的选项。

表 3.6. glance.store.rbd.store

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
<code>rados_connect_timeout = 0</code>	整数值	<p>连接到 Ceph 集群的超时值。</p> <p>此配置选项在连接到 Ceph 集群时所用的超时值（以秒为单位）。它会设置在关闭连接前等待 glance-api 的时间。这可防止在连接到 RBD 时出现 glance-api 挂起。如果此选项的值设为小于或等于 0，则不设置超时，并且使用默认的 librados 值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何整数值 <p>相关选项：</p> <ul style="list-style-type: none"> None
<code>`rd_store_ceph_conf = `</code>	字符串值	<p>Ceph 配置文件路径。</p> <p>此配置选项指定要使用的 Ceph 配置文件的路径。如果此选项的值未由用户设置或设置为空字符串，librados 将按顺序搜索默认的 Ceph 配置文件位置来读取标准 ceph.conf 文件。详情请参阅 Ceph 文档。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>如果使用 Cephx 身份验证，此文件应该在 client.<USER> 部分中包含对正确密钥环的引用</p> </div> </div> <p>注意 2：如果您将此选项留空（默认），使用的实际 Ceph 配置文件可能会根据使用 librados 的版本而有所不同。如果您知道哪个配置文件生效非常重要，您可以在此处使用这个选项指定该文件。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 配置文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> rd_store_user

配置选项 = 默认值	类型	Description
rdp_store_chunk_size = 8	整数值	<p>大小（以 MB 为单位）将 RADOS 镜像块到其中。</p> <p>提供代表 Glance 镜像的大小（以 MB 为单位）的整数值。默认块大小为 8MB。为获得最佳性能，该值应该是 2 的指数。</p> <p>当 Ceph 的 RBD 对象存储系统用作存储 Glance 镜像的存储后端时，镜像将使用此选项集中到设置的大小的对象中。然后，这些块的对象存储在用于 Glance 的分布式块数据存储中。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> None
rdp_store_pool = images	字符串值	<p>存储镜像的 RADOS 池。</p> <p>当 RBD 用作存储 Glance 镜像的存储后端时，镜像通过将对象（镜像区块）的逻辑分组存储到池中。每个池使用它可以包含的放置组数量来定义。使用的默认池是 <i>images</i>。</p> <p>有关 RBD 存储后端的更多信息，请参阅： http://ceph.com/planet/how-data-is-stored-in-ceph-cluster/</p> <p>可能的值：</p> <ul style="list-style-type: none"> 有效池名称 <p>相关选项：</p> <ul style="list-style-type: none"> None

配置选项 = 默认值	类型	Description
rbd_store_user = None	字符串值	<p>RADOS 用户以以下身份进行身份验证：</p> <p>此配置选项在 RADOS 用户进行身份验证，以如下方式进行身份验证：这仅在启用 RADOS 身份验证时且仅在用户使用 Cephx 身份验证时才需要。如果此选项的值未由用户设置或设置为 None，则会选择一个默认值，该值将基于 rbd_store_ceph_conf 中的 client. 部分。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的 RADOS 用户 <p>相关选项：</p> <ul style="list-style-type: none"> ● rbd_store_ceph_conf
rbd_thin_provisioning = False	布尔值	<p>在这个后端中启用或不精简置备。</p> <p>此配置选项允许在 RBD 后端上实际写入 null 字节序列的功能，Ceph 将自动显示的漏洞被 Ceph 解释为 null 字节，且不会实际消耗您的存储。启用此功能也会加快镜像上传速度，并节省后端中的空间，因为不会通过网络发送 null 字节序列。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● None

3.1.8. glance.store.s3.store

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[glance.store.s3.store]` 组下可用的选项。

表 3.7. glance.store.s3.store

配置选项 = 默认值	类型	描述
------------	----	----

配置选项 = 默认值	类型	描述
s3_store_access_key = None	字符串值	<p>S3 查询令牌访问密钥。</p> <p>此配置选项使用访问密钥来与 Amazon S3 或 S3 兼容存储服务器进行身份验证。使用 S3 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串值，这是具有适当特权的用户的访问密钥 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_host s3_store_secret_key
s3_store_bucket = None	字符串值	<p>用于存储 Glance 数据的 S3 存储桶。</p> <p>此配置选项指定 glance 镜像存储在 S3 中的位置。如果 s3_store_create_bucket_on_put 设为 true，它将自动创建，即使存储桶不存在。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_create_bucket_on_put s3_store_bucket_url_format
s3_store_bucket_url_format = auto	字符串值	<p>用于确定对象的 S3 调用格式。</p> <p>此配置选项使用访问模型，用于在 S3 存储桶中指定对象地址。</p> <p>注意：在 路径- 样式中，对象的端点类似于 https://s3.amazonaws.com/bucket/example.img。在 虚拟- 样式中，对象的端点类似于 https://bucket.s3.amazonaws.com/example.img。如果您没有在存储桶名称中遵循 DNS 命名惯例，您可以使用路径风格获取对象，但不能以虚拟样式获取对象。</p> <p>可能的值：</p> <ul style="list-style-type: none"> auto、虚拟 或 path 的任何字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_bucket

配置选项 = 默认值	类型	描述
s3_store_create_bucket_on_put = False	布尔值	<p>确定 S3 是否应该创建新存储桶。</p> <p>此配置选项使用布尔值来指示 Glance 是否应该向 S3 创建新存储桶（如果不存在）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何布尔值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
s3_store_host = None	字符串值	<p>S3 服务器侦听的主机。</p> <p>此配置选项设置 S3 或 S3 兼容存储服务器的主机。使用 S3 存储后端时需要这个选项。主机可以包含 DNS 名称（如 s3.amazonaws.com、my-object-storage.com）或 IP 地址(127.0.0.1)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的 DNS 名称 ● 有效的 IPv4 地址 <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_access_key ● s3_store_secret_key
s3_store_large_object_chunk_size = 10	整数值	<p>在上传部分时，S3 应该使用什么多部分上传部分大小（以 MB 为单位）。</p> <p>这个配置选项使用镜像分割大小（以 MB 为单位）进行多部分上传。</p> <p>注意：您只能分成 10,000 个镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值（必须大于或等于 5M） <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_large_object_size ● s3_store_thread_pools

配置选项 = 默认值	类型	描述
s3_store_large_object_size = 100	整数值	<p>S3 启动镜像文件的大小应该以 MB 为单位，并在 S3 中进行多部分上传。</p> <p>此配置选项采用以 MB 为单位的阈值，以确定是否将镜像上传到 S3，还是将其分割（多部分上传）。</p> <p>注意：您只能分成 10,000 个镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_large_object_chunk_size s3_store_thread_pools
s3_store_secret_key = None	字符串值	<p>S3 查询令牌 secret 密钥。</p> <p>此配置选项使用 secret 密钥来与 Amazon S3 或 S3 兼容存储服务器进行身份验证。使用 S3 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串值，它是与使用 s3_store_host 选项指定的访问密钥对应的 secret 键 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_host s3_store_access_key
s3_store_thread_pools = 10	整数值	<p>在 S3 中执行多部分上传的线程池数量。</p> <p>此配置选项在执行多部分上传时取线程池的数量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_large_object_size s3_store_large_object_chunk_size

3.1.9. glance.store.swift.store

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[glance.store.store]` 组下可用的选项。


表 3.8. glance.store.swift.store

配置选项 = 默认值	类型	Description
default_swift_reference = ref1	字符串值	<p>引用默认 Swift 帐户/备份存储参数。</p> <p>提供代表使用 swift 帐户/反馈存储所需默认参数集合的引用的字符串值。此配置选项的默认引用值为 <i>ref1</i>。此配置选项解引用参数，并有助于每次添加新镜像时 Swift 存储后端中的镜像存储。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_buffer_on_upload = False	布尔值	<p>上传到 Swift 前缓冲区镜像段。</p> <p>提供布尔值，以指示 Glance 是否应该在上传到 swift 时将镜像数据缓冲到磁盘。这可让 Glance 恢复出错时上传。</p> <p>注意：当启用这个选项时，应该非常小心，因为这会增加 API 节点上的磁盘用量。请注意，根据文件系统的配置方式，用于缓冲的磁盘空间可能会减少 glance 镜像缓存可用的实际磁盘空间。磁盘利用率将根据以下等情况设置： (swift_store_large_object_chunk_size * workers * 1000)</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_upload_buffer_dir

配置选项 = 默认值	类型	Description
swift_store_admin_tenants = []	列表值	<p>将被授予 admin 访问权限的租户列表。</p> <p>这是租户列表，将在多租户模式下授予 Glance 创建的所有 Swift 容器的读取/写入访问权限。默认值为空列表。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 以逗号分隔的字符串列表，代表 Keystone 项目/租户的 UUID <p>相关选项：</p> <ul style="list-style-type: none"> None
swift_store_auth_address = None	字符串值	Swift 身份验证服务侦听的端口。
swift_store_auth_insecure = False	布尔值	<p>设置服务器证书验证。</p> <p>此布尔值决定了是否验证服务器证书。如果此选项设置为 True，则 swiftclient 在验证时不会检查有效的 SSL 证书。如果选项设为 False，则使用默认 CA 信任存储进行验证。</p> <p>可能的值：</p> <ul style="list-style-type: none"> True False <p>相关选项：</p> <ul style="list-style-type: none"> swift_store_cacert
swift_store_auth_version = 2	字符串值	要使用的身份验证服务的版本。对于 keystone 有效版本是 2 和 3，对于 swauth 和 rackspace 为 1（已弃用）。
swift_store_cacert = None	字符串值	<p>CA 捆绑包文件的路径。</p> <p>此配置选项可让操作员指定自定义证书颁发机构文件的路径，以便在连接到 Swift 时进行 SSL 验证。</p> <p>可能的值：</p> <ul style="list-style-type: none"> CA 文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> swift_store_auth_insecure

配置选项 = 默认值	类型	Description
<p>swift_store_config_file = None</p>	<p>字符串值</p>	<p>包含 swift 帐户配置的文件的路径。</p> <p>包括一个字符串值，代表配置文件的路径，该文件对配置的每个 Swift 帐户/反馈存储的引用。默认情况下，未指定文件路径，并且禁用自定义 Swift 引用。强烈建议配置这个选项，同时将 Swift 存储后端用于镜像存储，因为它避免存储数据库中的凭证。</p> <p> 注意</p> <p>如果将 swift_store_multi_tenant 设为 True，请不要配置此选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表 glance-api 节点上的绝对路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_multi_tenant

配置选项 = 默认值	类型	Description
<p>swift_store_container = glance</p>	<p>字符串值</p>	<p>用于存储多个容器的 images/name 前缀的单一容器名称</p> <p>当单个容器用于存储镜像时，此配置选项将指示 Glance 帐户内用于存储所有镜像的容器。当使用多个容器来存储镜像时，这将是所有容器的名称前缀。可以使用配置选项 swift_store_multiple_containers_seed 来控制单/multiple 容器的使用。</p> <p>使用多个容器时，容器将通过为此配置选项设置的值命名，并将镜像 UUID 的第一个 N chars 指定为以下划线分隔的后缀（其中 N 由 swift_store_multiple_containers_seed 指定）。</p> <p>示例：如果 seed 设为 3，swift_store_container = glance，则带有 UUID fdac39a1-bac5-4238-aba4-69bcc726e848 的一个镜像将被放入容器 glance_fda。在创建容器名称时包括在 UUID 中的横线不会计算在字符长度的限制中，因此，当 N= 10 时，容器名称可以为 glance_fdac39a1-ba.</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 如果使用单个容器，此配置选项可以是 Glance Swift 帐户中有效 swift 容器名称的任何字符串 ● 如果使用多个容器，此配置选项可以是任何字符串，只要它满足 Swift 强制执行的容器命名规则。swift_store_multiple_containers_seed 的值也应该考虑。 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_multiple_containers_seed ● swift_store_multi_tenant ● swift_store_create_container_on_put


配置选项 = 默认值	类型	Description
swift_store_create_container_on_put = False	布尔值	<p>上传镜像时，创建容器（如果不存在）。</p> <p>在上传镜像时，如果对应的容器不存在，则会创建它，因为此配置选项被设置为 True。默认情况下，不会创建它。这个行为适用于单一和多个容器模式。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_endpoint = None	字符串值	<p>用于 Swift 后端存储的 URL 端点。</p> <p>提供代表在 Swift 存储中存储的 Glance 镜像的 URL 端点的字符串值。默认情况下，不会设置端点，并使用 auth 返回的存储 URL。使用 swift_store_endpoint 设置端点会覆盖存储 URL，并用于 Glance 镜像存储。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>URL 应该包括到的路径，但不包括容器。对象的位置通过将容器和对象附加到配置的 URL 来获取。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表到 Swift 容器的有效 URL 路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
swift_store_endpoint_type = publicURL	字符串值	<p>Swift 服务的端点类型。</p> <p>此字符串值表示用于获取 Swift 端点的端点类型。端点类型决定了用户被允许执行的操作，例如，读取和写入存储。只有在 swift_store_auth_version 大于 1 时才会使用此设置。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● publicURL ● adminURL ● internalURL <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_endpoint
swift_store_expire_soon_interval = 60	整数值	<p>定义在当前令牌到期前可以请求新令牌的窗口大小的时间（以秒为单位）。</p> <p>通常，Swift 存储驱动程序会在当前令牌过期时获取新令牌，以确保继续访问 Swift。但是，如果令牌实时过期，一些 Swift 事务（如上传镜像段）可能无法正常恢复。</p> <p>因此，通过在当前令牌过期前获取新令牌，请确保令牌不会在尝试事务前过期或接近到期。默认情况下，Swift 存储驱动程序请求新令牌 60 秒或少于当前令牌过期前的请求。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 零 ● 正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_key = None	字符串值	<p>用于针对 Swift 身份验证服务进行身份验证的验证密钥。</p>


配置选项 = 默认值	类型	Description
swift_store_large_object_chunk_size = 200	整数值	<p>镜像数据时段的最大大小（以 MB 为单位）。</p> <p>当镜像数据被分割为上传大于 Swift 集群实施的限制的镜像时，镜像数据被分成不同的片段，它们的大小不大于该配置选项指定的大小。如需了解更多详细信息，请参阅 swift_store_large_object_size。</p> <p>例如：如果 swift_store_large_object_size 为 5GB，而 swift_store_large_object_chunk_size 为 1GB，则大小为 6.2GB 的镜像将被分为 7 段，前 6 个段的大小为 1GB，第七个段的大小为 0.2GB。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数，它小于或等于 Swift 集群强制的大型对象限制。 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_large_object_size
swift_store_large_object_size = 5120	整数值	<p>大小阈值（以 MB 为单位）之后，Glance 将开始分段镜像数据。</p> <p>Swift 对单个上传对象的大小具有上限。默认情况下，这是 5GB。要上传大于这个限制的对象，对象被分成多个与清单文件绑定的较小的对象。如需更多详情，请参阅 https://docs.openstack.org/swift/latest/overview_large_objects.html</p> <p>此配置选项指定 Swift 驱动程序开始将镜像数据段到多个更小文件的大小阈值。目前，Swift 驱动程序只支持创建动态大对象。</p> <div data-bbox="815 1464 922 1599" style="display: inline-block; vertical-align: middle;">  </div> <p style="margin-left: 20px;">注意</p> <p style="margin-left: 20px;">这应该通过考虑 Swift 集群强制的大型对象限制来设置。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数，它小于或等于 Swift 集群强制的大型对象限制。 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_large_object_chunk_size

配置选项 = 默认值	类型	Description
swift_store_multi_tenant = False	布尔值	<p>将镜像存储在租户的 Swift 帐户中。</p> <p>这将启用多租户存储模式，使 Glance 镜像存储在租户特定 Swift 帐户中。如果禁用，Glance 会将所有镜像存储在自己的帐户中。有关多租户存储的详情，请参考 https://wiki.openstack.org/wiki/GlanceSwiftTenantSpecificStorage</p> <p> 注意</p> <p>如果使用多租户 swift 存储，请确保不要使用 <code>swift_store_config_file</code> 选项设置 swift 配置文件。</p> <p>可能的值：</p> <ul style="list-style-type: none">• True• False <p>相关选项：</p> <ul style="list-style-type: none">• <code>swift_store_config_file</code>

配置选项 = 默认值	类型	Description
swift_store_multiple_containers_seed = 0	整数值	<p>指出用于存储镜像的容器数量。</p> <p>在使用单租户存储时，镜像可以存储在一个或多个容器中。当设置为 0 时，所有镜像将存储在一个容器中。当设置为 1 到 32 之间的整数值时，将使用多个容器来存储镜像。此配置选项将确定创建多少个容器。使用的容器总数等于 16^N，因此如果该配置选项设为 2，则使用 $16^2=256$ 容器来存储镜像。</p> <p>有关命名规则的详情，请参阅 swift_store_container。有关使用多个容器的详情，请参考 https://specs.openstack.org/openstack/glance-specs/specs/kilo/swift-store-multiple-containers.html</p> <p> 注意</p> <p>这仅在禁用 <code>swift_store_multi_tenant</code> 时使用。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 小于或等于 32 的非负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_container ● swift_store_multi_tenant ● swift_store_create_container_on_put

配置选项 = 默认值	类型	Description
swift_store_region = None	字符串值	<p>Glance 使用的 Swift 端点区域。</p> <p>提供代表 Swift 区域的字符串值，其中 Glance 可以连接到该区域进行镜像存储。默认情况下，没有设置区域。</p> <p>当 Glance 使用 Swift 作为存储后端存储具有多个端点的特定租户的镜像时，使用 swift_store_region 设置 Swift 区域将允许 Glance 连接到指定区域中的 Swift，而不是单一区域连接。</p> <p>可以为单租户和多租户存储配置此选项。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>使用 swift_store_region 设置区域是特定于租户的，只有在 租户在不同区域上有多个端点时才需要。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表有效 Swift 区域的字符串值。 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_retry_get_count = 0	整数值	<p>在请求失败前，将重试 Swift 下载的次数。</p> <p>提供一个整数值，代表镜像下载的次数，然后才能出错。默认值为 0（在失败的镜像下载时不重试）。当设置为正整数值时，swift_store_retry_get_count 确保在发送错误消息前下载失败时尝试多次下载。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 零 ● 正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
swift_store_service_type = object-store	字符串值	<p>要使用的 Swift 服务类型。</p> <p>提供代表在使用 Swift 后端存储期间存储镜像的服务类型的字符串值。默认服务类型设置为 object-store。</p> <div data-bbox="815 398 922 629" style="float: left; margin-right: 10px;">  </div> <p>注意</p> <p>如果 swift_store_auth_version 设为 2，则此配置选项的值需要是 object-store。如果使用较新版本的 Keystone 或不同的身份验证方案，则可以修改此选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表 Swift 存储的有效服务类型的字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_ssl_compression = True	布尔值	<p>HTTPS Swift 请求的 SSL 层压缩。</p> <p>提供布尔值，以确定是否压缩 SSL 层上镜像的 HTTPS Swift 请求。默认情况下启用压缩。</p> <p>当使用 Swift 作为 Glance 镜像存储的后端存储时，可以使用此选项设置 HTTPS Swift 请求的 SSL 层压缩。如果设置为 False，则禁用 HTTPS Swift 请求的 SSL 层压缩。禁用这个选项可能会提高已压缩格式的镜像的性能，如 qcow2。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
swift_store_use_trusts = True	布尔值	<p>对多租户 Swift 存储使用 trusts。</p> <p>此选项指示 Swift 存储在使用多租户存储时为每个 add/get 请求创建一个信任。使用信任允许 Swift 存储避免在上传或下载数据期间由身份验证令牌过期的问题。</p> <p>默认情况下，swift_store_use_trusts 被设置为 True（启用使用信任）。如果设置为 False，则用户令牌用于 Swift 连接，从而消除了创建信任的开销。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>只有在 swift_store_multi_tenant 被设置为 True 时，才会考虑此选项</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_multi_tenant
swift_store_user = None	字符串值	对 Swift 身份验证服务进行身份验证的用户。

配置选项 = 默认值	类型	Description
swift_upload_buffer_dir = None	字符串值	<p>上传到 Swift 前用于缓冲镜像片段的目录。</p> <p>提供一个字符串值，代表 glance 节点上目录的绝对路径，在镜像片段上传到 swift 之前会短暂缓冲镜像段。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 这只有在配置选项 swift_buffer_on_upload 设置为 True 时才需要。 ● 应置备此目录，请记住 swift_store_large_object_chunk_size 以及给定 glance 节点可以同时上传的最大镜像数。 <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表绝对路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_buffer_on_upload ● swift_store_large_object_chunk_size

3.1.10. glance.store.vmware_datastore.store

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[glance.store.vmware_datastore.store]` 组下可用的选项。

表 3.9. glance.store.vmware_datastore.store

配置选项 = 默认值	类型	Description
vmware_api_retry_count = 10	整数值	<p>VMware API 重试次数。</p> <p>此配置选项指定连接相关问题或服务器 API 调用过载时必须重试 VMware ESX/VC 服务器 API 的次数。无法指定 <i>retry forever</i>。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
vmware_ca_file = None	字符串值	<p>CA 捆绑包文件的绝对路径。</p> <p>此配置选项可让操作员使用自定义 Certificate 授权机构文件来验证 ESX/vCenter 证书。</p> <p>如果设置了这个选项，则忽略 "vmware_insecure" 选项，指定的 CA 文件将用于验证 ESX/vCenter 服务器证书，并建立与服务器的安全连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何是 CA 文件的有效绝对路径的字符串 <p>相关选项：</p> <ul style="list-style-type: none"> vmware_insecure
vmware_datastores = None	多值	<p>用于存储镜像的数据存储。</p> <p>此配置选项指定镜像存储在 VMWare 存储后端中的数据存储。这个选项可多次指定，以指定多个数据存储。数据存储名称应在其数据中心路径后指定，用 ":" 分隔。可以在数据存储名称后赋予可选权重，再次通过 ":" 分隔来指定优先级。因此，所需格式将变为 <datacenter_path>:<datastore_name>:<optional_weight>。</p> <p>在添加镜像时，会选择权重最高的数据存储，除非已知镜像大小时没有足够的可用空间。如果未指定权重，则假定为 0，目录将被视为最后选择。如果多个数据存储具有相同的权重，则会选择一个带有最多可用空间的权重。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何格式的字符串：<datacenter_path>:<datastore_name>:<optional_weight> <p>相关选项：* None</p>

配置选项 = 默认值	类型	Description
vmware_insecure = False	布尔值	<p>设置 ESX/vCenter 服务器证书验证。</p> <p>此配置选项使用布尔值来确定是否验证 ESX/vCenter 服务器证书。如果此选项设为 True，则不会验证 ESX/vCenter 服务器证书。如果此选项设置为 False，则使用默认 CA 信任存储进行验证。</p> <p>如果设置了 "vmware_ca_file" 选项，则忽略这个选项。在这种情况下，将使用使用 "vmware_ca_file" 选项指定的文件来验证 ESX/vCenter 服务器证书。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • vmware_ca_file
vmware_server_host = None	主机地址值	<p>ESX/ESXi 或 vCenter Server 目标系统的地址。</p> <p>此配置选项设置 ESX/ESXi 或 vCenter Server 目标系统的地址。使用 VMware 存储后端时需要这个选项。地址可以包含 IP 地址(127.0.0.1)或 DNS 名称 (www.my-domain.com)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 有效的 IPv4 或 IPv6 地址 • 有效的 DNS 名称 <p>相关选项：</p> <ul style="list-style-type: none"> • vmware_server_username • vmware_server_password

配置选项 = 默认值	类型	Description
vmware_server_password = None	字符串值	<p>服务器密码。</p> <p>此配置选项使用密码通过 VMware ESX/ESXi 或 vCenter 服务器进行身份验证。使用 VMware 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串，与使用 "vmware_server_username" 选项指定的用户名对应的密码 <p>相关选项：</p> <ul style="list-style-type: none"> vmware_server_host vmware_server_username
vmware_server_username = None	字符串值	<p>服务器用户名。</p> <p>此配置选项使用用户名来通过 VMware ESX/ESXi 或 vCenter 服务器进行身份验证。使用 VMware 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串，这是具有适当权限的用户的用户名 <p>相关选项：</p> <ul style="list-style-type: none"> vmware_server_host vmware_server_password
vmware_store_image_dir = /openstack_glance	字符串值	<p>glance 镜像要存储在数据存储中的目录。</p> <p>此配置选项指定 glance 镜像存储在 VMware 数据存储中的目录的路径。如果没有设置这个选项，则存储 glance 镜像的默认目录为 openstack_glance。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何是到目录的有效路径的字符串 <p>相关选项：</p> <ul style="list-style-type: none"> None

配置选项 = 默认值	类型	Description
vmware_task_poll_interval = 5	整数值	<p>用于轮询在 VMware ESX/VC 服务器上调用的远程任务的间隔（以秒为单位）。</p> <p>此配置选项需要休眠时间（以秒为单位）轮询 on-going async 任务，作为 VMWare ESX/VC 服务器 API 调用的一部分。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

3.1.11. glance_store

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[glance_store]` 组下可用的选项。

表 3.10. glance_store

配置选项 = 默认值	类型	Description
cinder_api_insecure = False	布尔值	<p>允许对 cinder 执行不安全的 SSL 请求。</p> <p>如果此选项被设置为 True，则使用 cinder_ca_certificates_file 选项指定的 CA 证书文件验证 HTTPS 端点连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_ca_certificates_file


配置选项 = 默认值	类型	Description
cinder_ca_certificate = None	字符串值	<p>用于 Cinder 客户端请求的 CA 证书文件的位置。</p> <p>指定的 CA 证书文件（如果设置）用于通过 HTTPS 端点验证 Cinder 连接。如果端点是 HTTP，则忽略这个值。cinder_api_insecure 必须设置为 True 以启用验证。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● ca 证书文件的路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_api_insecure
cinder_catalog_info = volumev3::publicURL	字符串值	<p>在服务目录中查找 cinder 时匹配的信息。</p> <p>如果没有设置 cinder_endpoint_template，并且任何 cinder_store_auth_address，cinder_store_user_name，cinder_store_project_name，cinder_store_password 没有设置，cinder 存储将使用此信息从当前上下文中的服务目录中查找 Cinder 端点。如果设置，将考虑 cinder_os_region_name 来获取适当的端点。</p> <p>服务目录可以通过 openstack catalog list 命令列出。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 应指定以下形式的字符串：< service_type>:<service_name>:<interface> At least service_type 和 interface。 service_name 可以省略。 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_os_region_name ● cinder_endpoint_template ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password

配置选项 = 默认值	类型	Description
cinder_endpoint_template = None	字符串值	<p>使用 cinder 端点的模板覆盖服务目录查找。</p> <p>设置此选项时，这个值用于生成 cinder 端点，而不是从服务目录查找。如果指定了 cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, 和 cinder_store_password, 则忽略这个值。</p> <p>如果设置了此配置选项，则忽略 cinder_catalog_info。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● cinder 端点的 URL 模板字符串，其中 %%(tenant) 替换为当前的租户（项目）名称。例如： http://cinder.openstack.example.org/v2/%%(tenant)s <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password ● cinder_catalog_info
cinder_enforce_multipath = False	布尔值	<p>如果设置为 True，则当 multipathd 未运行时，镜像传输的卷附加将中止。否则，它将回退到单一路径。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True 或 False <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_use_multipath

配置选项 = 默认值	类型	Description
cinder_http_retries = 3	整数值	<p>失败 http 调用的 cinderclient 重试次数。</p> <p>当调用被任何错误失败时，cinderclient 会在睡眠几秒钟后重试调用。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
cinder_mount_point_base = /var/lib/glance/mnt	字符串值	<p>NFS 卷挂载在 glance 节点上的目录。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表挂载点的绝对路径的字符串。
cinder_os_region_name = None	字符串值	<p>从服务目录查找 cinder 服务的区域名称。</p> <p>这仅在使用 cinder_catalog_info 来确定端点时使用。如果设置，则此节点的 cinder 端点的查找将过滤到指定的区域。当目录中列出了多个区域时，它很有用。如果没有设置，则会从每个区域查找端点。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 是有效区域名称的字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_catalog_info
cinder_state_transition_timeout = 300	整数值	<p>等待 cinder 卷转换完成的时间（以秒为单位）。</p> <p>当创建、删除或附加到 glance 节点以读/写卷数据时，卷的状态将改变。例如，新创建的卷状态会在 创建 过程完成后变为 available。这将指定等待状态更改的最长时间。如果在等待时发生超时，或者状态更改为非预期的值（如 错误），则镜像创建会失败。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
cinder_store_auth_address = None	字符串值	<p>cinder 身份验证服务正在侦听的地址。</p> <p>当指定所有 cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, 和 cinder_store_password 选项时, 指定的值始终用于身份验证。这有助于通过将镜像服务存储在特定于镜像服务的项目/租户中, 从用户隐藏镜像卷。它还允许用户在 glance 的 ACL 控制下与其他项目共享镜像卷。</p> <p>如果没有设置任何一个选项, 则 cinder 端点会从服务目录中查找, 并使用当前上下文的用户和项目。</p> <p>可能的值 :</p> <ul style="list-style-type: none"> 有效的身份验证服务地址, 例如 : http://openstack.example.org/identity/v2.0 <p>相关选项 :</p> <ul style="list-style-type: none"> cinder_store_user_name cinder_store_password cinder_store_project_name
cinder_store_password = None	字符串值	<p>用于针对 Cinder 进行身份验证的用户密码。</p> <p>这必须与以下所有相关选项一起使用。如果未指定其中任何一个, 则使用当前上下文的用户。</p> <p>可能的值 :</p> <ul style="list-style-type: none"> 由 cinder_store_user_name 指定的用户的有效密码 <p>相关选项 :</p> <ul style="list-style-type: none"> cinder_store_auth_address cinder_store_user_name cinder_store_project_name

配置选项 = 默认值	类型	Description
cinder_store_project_name = None	字符串值	<p>镜像卷的项目名称存储在 cinder 中。</p> <p>如果没有设置此配置选项，则使用当前上下文中的项目。</p> <p>这必须与以下所有相关选项一起使用。如果未指定其中任何一个，则使用当前上下文的项目。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效项目名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_password
cinder_store_user_name = None	字符串值	<p>要针对 cinder 进行身份验证的用户名。</p> <p>这必须与以下所有相关选项一起使用。如果未指定其中任何一个，则使用当前上下文的用户。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的用户名 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_password ● cinder_store_project_name
cinder_use_multipath = False	布尔值	<p>在部署中支持或不支持识别 mutipath 的标记。</p> <p>如果不支持多路径，则将其设置为 False。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True 或 False <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_enforce_multipath

配置选项 = 默认值	类型	Description
<p>cinder_volume_type = None</p>	<p>字符串值</p>	<p>在 cinder 中用于创建卷的卷类型。</p> <p>有些 Cinder 后端可以有几个卷类型来优化存储使用。添加此选项可让操作器在 cinder 中选择特定卷类型，该类型可以针对镜像进行了优化。</p> <p>如果没有设置，则 cinder 配置中指定的默认卷类型将用于创建卷。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 来自 cinder 的有效卷类型 <p>相关选项：</p> <ul style="list-style-type: none"> ● None <div data-bbox="815 792 922 1055" style="border: 1px solid black; padding: 5px; width: fit-content;">  </div> <p>注意</p> <p>您不能使用与 NFS 后端关联的加密 volume_type。当 glance_store 试图写入或访问存储在该卷中的镜像数据时，存储在 NFS 后端上的加密卷将引发异常。请参考 Cinder 管理员确定适当的 volume_type。</p>
<p>default_backend = None</p>	<p>字符串值</p>	<p>存储数据的默认后端的存储标识符。</p> <p>该值必须定义为 DEFAULT 配置组中 enabled_backends 配置选项定义的字典中的其中一个键。</p> <p>如果没有为这个选项定义值：</p> <ul style="list-style-type: none"> ● 使用的服务可能会拒绝启动 ● 没有指定特定后端的 store_add 调用将引发 glance_store.exceptions.UnknownScheme 异常 <p>相关选项：</p> <ul style="list-style-type: none"> ● enabled_backends

配置选项 = 默认值	类型	Description
default_store = file	字符串值	<p>用于存储镜像的默认方案。</p> <p>提供代表用于存储镜像的默认方案的字符串值。如果没有设置，Glance 使用 file 作为默认方案来使用文件存储镜像。</p> <div data-bbox="815 439 922 602" style="float: left; margin-right: 10px;">  </div> <p>注意</p> <p>为这个配置选项提供的值必须是使用 stores 配置选项注册的存储的有效方案。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● file ● filesystem ● http ● https ● swift ● swift+http ● swift+https ● swift+config ● rbd ● cinder ● vsphere ● s3 <p>相关选项：</p> <ul style="list-style-type: none"> ● 存储 <p>弃用自： Rocky</p> <p>reason: 这个选项根据新配置选项 "default_backend" 弃用，它的行为与 "default_store" 配置选项类似。</p> <p>计划在 U 开发周期中移除此选项。</p>

配置选项 = 默认值	类型	Description
default_swift_reference = ref1	字符串值	<p>引用默认 Swift 帐户/备份存储参数。</p> <p>提供代表使用 swift 帐户/反馈存储所需默认参数集合的引用的字符串值。此配置选项的默认引用值为 <i>ref1</i>。此配置选项解引用参数，并有助于每次添加新镜像时 Swift 存储后端中的镜像存储。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
filesystem_store_chunk_size = 65536	整数值	<p>块大小，以字节为单位。</p> <p>读取或写入镜像文件时使用的块大小。增加这个值可以提高吞吐量，但在处理大量请求时，它也可能略微增加内存用量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
<p>filesystem_store_datadir = /var/lib/glance/images</p>	<p>字符串值</p>	<p>文件系统后端存储写入镜像的目录。</p> <p>启动时，Glance 会创建该目录（如果不存在），并验证在其下运行 glance-api 的用户的写入访问权限。如果写入访问不可用，则引发 BadStoreConfiguration 异常，并且文件系统存储可能无法添加新镜像。</p> <div data-bbox="815 510 922 824" style="border: 1px solid gray; padding: 5px; width: fit-content;">  </div> <p>注意</p> <p>此目录仅在将文件系统存储用作存储后端时使用。filesystem_store_datadir 或 filesystem_store_datadirs 选项必须在 glance-api.conf 中指定。如果指定了这两个选项，则会引发 BadStoreConfiguration，并且文件系统存储可能无法添加新镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 到目录的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● filesystem_store_datadirs ● filesystem_store_file_perm

配置选项 = 默认值	类型	Description
filesystem_store_datadirs = None	多值	<p>文件系统后端存储写入镜像的目录及其优先级列表。</p> <p>文件系统存储可以配置为将镜像存储在多个目录中，而不是使用由 filesystem_store_datadir 配置选项指定的单个目录。使用多个目录时，可以为每个目录赋予一个可选优先级，以指定应使用它们的首选项顺序。priority 是一个整数，它将与目录路径连接，冒号越高，代表优先级更高。当两个目录具有相同的优先级时，会使用具有最多可用空间的目录。如果没有指定优先级，则默认为零。</p> <p>有关使用多个存储目录配置文件系统存储的更多信息，请参阅 https://docs.openstack.org/glance/latest/configuration/configuring.html</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 40px; height: 40px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, #ccc 2px, #ccc 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>此目录仅在将文件系统存储用作存储后端时使用。filesystem_store_datadir 或 filesystem_store_datadirs 选项必须在 glance-api.conf 中指定。如果指定了这两个选项，则会引发 BadStoreConfiguration，并且文件系统存储可能无法添加新镜像。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 以下格式的字符串列表： <ul style="list-style-type: none"> ○ <a valid directory path>:<optional integer priority> <p>相关选项：</p> <ul style="list-style-type: none"> ● filesystem_store_datadir ● filesystem_store_file_perm

配置选项 = 默认值	类型	Description
filesystem_store_file_permission = 0	整数值	<p>镜像文件的文件访问权限。</p> <p>为镜像数据设置预期的文件访问权限。这提供了一种方式，可以让其他服务（如 Nova）直接使用文件系统存储中的镜像。运行要被授予访问权限的服务的用户，可以成为拥有创建文件的组的成员。为这个配置选项分配一个小于或等于 0 的值表示没有对默认权限进行任何更改。这个值将被解码为八进制数。</p> <p>如需更多信息，请参阅文档 https://docs.openstack.org/glance/latest/configuration/configuring.html</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的文件访问权限 ● 零 ● 任何负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
filesystem_store_metadata_file = None	字符串值	<p>文件系统存储元数据文件。</p> <p>包含与文件系统存储关联的任何位置返回元数据的文件路径。设置此选项后，它仅用于之后创建的新镜像 - 以前现有镜像不受影响。</p> <p>该文件必须包含有效的 JSON 对象。该对象应包含键 id 和 mountpoint。两个键的值都应该是一个字符串。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 存储元数据文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
filesystem_thin_provisioning = False	布尔值	<p>在这个后端中启用或不精简置备。</p> <p>这个配置选项启用了在文件系统中没有真正写入 null 字节序列的功能，文件系统会自动解释为 null 字节，且不会实际消耗您的存储。启用此功能也会加快镜像上传速度，并节省后端中的空间，因为不会通过网络发送 null 字节序列。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • None
http_proxy_information = {}	dict 值	<p>用于连接到远程服务器的 http/https 代理信息。</p> <p>这个配置选项指定应该用来连接到远程服务器的 http/https 代理信息。代理信息应该是方案和代理的键值对，如 http:10.0.0.1:3128。您还可以通过使用逗号分隔键值对（如 http:10.0.0.1:3128, https:10.0.0.1:1080）来指定多个方案的代理。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 如前文所述，以逗号分隔的 scheme:proxy 对列表 <p>相关选项：</p> <ul style="list-style-type: none"> • None
https_ca_certificates_file = None	字符串值	<p>CA 捆绑包文件的路径。</p> <p>此配置选项可让操作员使用自定义证书颁发机构文件来验证远程服务器证书。如果设置了这个选项，则忽略 https_insecure 选项，指定的 CA 文件将用于验证服务器证书并建立到服务器的安全连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • CA 文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> • https_insecure

配置选项 = 默认值	类型	Description
https_insecure = True	布尔值	<p>设置远程服务器证书的验证。</p> <p>此配置选项使用布尔值来确定是否验证远程服务器证书。如果设置为 True，则不会验证远程服务器证书。如果选项设为 False，则使用默认 CA 信任存储进行验证。</p> <p>如果设置了 https_ca_certificates_file，则忽略这个选项。然后，使用 https_ca_certificates_file 选项指定的文件验证远程服务器证书。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • https_ca_certificates_file
rados_connect_timeout = 0	整数值	<p>连接到 Ceph 集群的超时值。</p> <p>此配置选项在连接到 Ceph 集群时所用的超时值（以秒为单位）。它会设置在关闭连接前等待 glance-api 的时间。这可防止在连接到 RBD 时出现 glance-api 挂起。如果此选项的值设为小于或等于 0，则不设置超时，并且使用默认的 librados 值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 任何整数值 <p>相关选项：</p> <ul style="list-style-type: none"> • None

配置选项 = 默认值	类型	Description
`rbd_store_ceph_conf = `	字符串值	<p>Ceph 配置文件路径。</p> <p>此配置选项指定要使用的 Ceph 配置文件的路径。如果此选项的值未由用户设置或设置为空字符串，librados 将按顺序搜索默认的 Ceph 配置文件位置来读取标准 ceph.conf 文件。详情请参阅 Ceph 文档。</p> <div data-bbox="815 472 922 636" style="float: left; margin-right: 10px;"> </div> <p>注意</p> <p>如果使用 Cephx 身份验证，此文件应该在 client.<USER> 部分中包含对正确密钥环的引用</p> <p>注意 2：如果您将此选项留空（默认），使用的实际 Ceph 配置文件可能会根据使用 librados 的版本而有所不同。如果您知道哪个配置文件生效非常重要，您可以在此处使用这个选项指定该文件。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 配置文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● rbd_store_user
rbd_store_chunk_size = 8	整数值	<p>大小（以 MB 为单位）将 RADOS 镜像块到其中。</p> <p>提供代表 Glance 镜像的大小（以 MB 为单位）的整数值。默认块大小为 8MB。为获得最佳性能，该值应该是 2 的指数。</p> <p>当 Ceph 的 RBD 对象存储系统用作存储 Glance 镜像的存储后端时，镜像将使用此选项集中到设置的大小的对象中。然后，这些块的对象存储在用于 Glance 的分布式块数据存储中。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
rbd_store_pool = images	字符串值	<p>存储镜像的 RADOS 池。</p> <p>当 RBD 用作存储 Glance 镜像的存储后端时，镜像通过将对象（镜像区块）的逻辑分组存储到 池中。每个池使用它可以包含的放置组数量来定义。使用的默认池是 <i>images</i>。</p> <p>有关 RBD 存储后端的更多信息，请参阅： http://ceph.com/planet/how-data-is-stored-in-ceph-cluster/</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效池名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
rbd_store_user = None	字符串值	<p>RADOS 用户以以下身份进行身份验证：</p> <p>此配置选项在 RADOS 用户进行身份验证，以如下方式进行身份验证：这仅在启用 RADOS 身份验证时且仅在用户使用 Cephx 身份验证时才需要。如果此选项的值未由用户设置或设置为 None，则会选择一个默认值，该值将基于 <code>rbd_store_ceph_conf</code> 中的 <code>client.</code> 部分。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的 RADOS 用户 <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>rbd_store_ceph_conf</code>

配置选项 = 默认值	类型	Description
rdp_thin_provisioning = False	布尔值	<p>在这个后端中启用或不精简置备。</p> <p>此配置选项允许在 RBD 后端上实际写入 null 字节序列的功能，Ceph 将自动显示的漏洞被 Ceph 解释为 null 字节，且不会实际消耗您的存储。启用此功能也会加快镜像上传速度，并节省后端中的空间，因为不会通过网络发送 null 字节序列。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • None
rootwrap_config = /etc/glance/rootwrap.conf	字符串值	<p>用于以 root 身份运行命令的 rootwrap 配置文件的路径。</p> <p>cinder 存储需要 root 特权才能运行镜像卷（用于连接 iSCSI/FC 卷并读取/写入卷数据等）。配置文件应允许 cinder 存储和 os-brick 库所需的命令。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • rootwrap 配置文件的路径 <p>相关选项：</p> <ul style="list-style-type: none"> • None
s3_store_access_key = None	字符串值	<p>S3 查询令牌访问密钥。</p> <p>此配置选项使用访问密钥来与 Amazon S3 或 S3 兼容存储服务器进行身份验证。使用 S3 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 任何字符串值，这是具有适当特权的用户的访问密钥 <p>相关选项：</p> <ul style="list-style-type: none"> • s3_store_host • s3_store_secret_key

配置选项 = 默认值	类型	Description
s3_store_bucket = None	字符串值	<p>用于存储 Glance 数据的 S3 存储桶。</p> <p>此配置选项指定 glance 镜像存储在 S3 中的位置。如果 s3_store_create_bucket_on_put 设为 true，它将自动创建，即使存储桶不存在。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_create_bucket_on_put s3_store_bucket_url_format
s3_store_bucket_url_format = auto	字符串值	<p>用于确定对象的 S3 调用格式。</p> <p>此配置选项使用访问模型，用于在 S3 存储桶中指定对象地址。</p> <p>注意：在 路径- 样式中，对象的端点类似于 https://s3.amazonaws.com/bucket/example.img。在 虚拟- 样式中，对象的端点类似于 https://bucket.s3.amazonaws.com/example.img。如果您没有在存储桶名称中遵循 DNS 命名惯例，您可以使用路径风格获取对象，但不能以虚拟样式获取对象。</p> <p>可能的值：</p> <ul style="list-style-type: none"> auto、虚拟 或 path 的任何字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_bucket
s3_store_create_bucket_on_put = False	布尔值	<p>确定 S3 是否应该创建新存储桶。</p> <p>此配置选项使用布尔值来指示 Glance 是否应该向 S3 创建新存储桶（如果不存在）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何布尔值 <p>相关选项：</p> <ul style="list-style-type: none"> None

配置选项 = 默认值	类型	Description
s3_store_host = None	字符串值	<p>S3 服务器侦听的主机。</p> <p>此配置选项设置 S3 或 S3 兼容存储服务器的主机。使用 S3 存储后端时需要这个选项。主机可以包含 DNS 名称（如 s3.amazonaws.com、my-object-storage.com）或 IP 地址(127.0.0.1)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的 DNS 名称 ● 有效的 IPv4 地址 <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_access_key ● s3_store_secret_key
s3_store_large_object_chunk_size = 10	整数值	<p>在上传部分时，S3 应该使用什么多部分上传部分大小（以 MB 为单位）。</p> <p>这个配置选项使用镜像分割大小（以 MB 为单位）进行多部分上传。</p> <p>注意：您只能分成 10,000 个镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值（必须大于或等于 5M） <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_large_object_size ● s3_store_thread_pools

配置选项 = 默认值	类型	Description
s3_store_large_object_size = 100	整数值	<p>S3 启动镜像文件的大小应该以 MB 为单位，并在 S3 中进行多部分上传。</p> <p>此配置选项采用以 MB 为单位的阈值，以确定是否将镜像上传到 S3，还是将其分割（多部分上传）。</p> <p>注意：您只能分成 10,000 个镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_large_object_chunk_size s3_store_thread_pools
s3_store_secret_key = None	字符串值	<p>S3 查询令牌 secret 密钥。</p> <p>此配置选项使用 secret 密钥来与 Amazon S3 或 S3 兼容存储服务器进行身份验证。使用 S3 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串值，它是与使用 s3_store_host 选项指定的访问密钥对应的 secret 键 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_host s3_store_access_key
s3_store_thread_pools = 10	整数值	<p>在 S3 中执行多部分上传的线程池数量。</p> <p>此配置选项在执行多部分上传时取线程池的数量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_large_object_size s3_store_large_object_chunk_size


配置选项 = 默认值	类型	Description
stores = ['file', 'http']	列表值	<p>启用 Glance 存储的列表。</p> <p>注册存储后端，以逗号分隔列表的形式用于存储磁盘镜像。为使用 Glance 存储磁盘镜像而启用的默认存储是 file 和 http。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个以逗号分隔的列表，其中包括： <ul style="list-style-type: none"> ○ file ○ http ○ swift ○ rbd ○ cinder ○ vmware ○ s3 <p>相关选项：</p> <ul style="list-style-type: none"> ● default_store <p>弃用自： Rocky</p> <p>reason: 这个选项根据新配置选项 "enabled_backends" 弃用，这有助于配置不同方案的多个后端存储。</p> <p>计划在 U 开发周期中移除此选项。</p>

配置选项 = 默认值	类型	Description
swift_buffer_on_upload = False	布尔值	<p>上传到 Swift 前缓冲区镜像段。</p> <p>提供布尔值，以指示 Glance 是否应该在上传到 swift 时将镜像数据缓冲到磁盘。这可让 Glance 恢复出错时上传。</p> <p>注意：当启用这个选项时，应该非常小心，因为这会增加 API 节点上的磁盘用量。请注意，根据文件系统的配置方式，用于缓冲的磁盘空间可能会减少 glance 镜像缓存可用的实际磁盘空间。磁盘利用率将根据以下等情况设置：</p> <p>(swift_store_large_object_chunk_size * workers * 1000)</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • swift_upload_buffer_dir
swift_store_admin_tenants = []	列表值	<p>将被授予 admin 访问权限的租户列表。</p> <p>这是租户列表，将在多租户模式下授予 Glance 创建的所有 Swift 容器的读取/写入访问权限。默认值为空列表。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 以逗号分隔的字符串列表，代表 Keystone 项目/租户的 UUID <p>相关选项：</p> <ul style="list-style-type: none"> • None
swift_store_auth_address = None	字符串值	Swift 身份验证服务侦听的端口。

配置选项 = 默认值	类型	Description
swift_store_auth_insecure = False	布尔值	<p>设置服务器证书验证。</p> <p>此布尔值决定了是否验证服务器证书。如果此选项设置为 True，则 swiftclient 在验证时不会检查有效的 SSL 证书。如果选项设为 False，则使用默认 CA 信任存储进行验证。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • swift_store_cacert
swift_store_auth_version = 2	字符串值	<p>要使用的身份验证服务的版本。对于 keystone 有效版本是 2 和 3，对于 swauth 和 rackspace 为 1（已弃用）。</p>
swift_store_cacert = None	字符串值	<p>CA 捆绑包文件的路径。</p> <p>此配置选项可让操作员指定自定义证书颁发机构文件的路径，以便在连接到 Swift 时进行 SSL 验证。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • CA 文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> • swift_store_auth_insecure

配置选项 = 默认值	类型	Description
swift_store_config_file = None	字符串值	<p>包含 swift 帐户配置的文件的路径。</p> <p>包括一个字符串值，代表配置文件的路径，该文件对配置的每个 Swift 帐户/反馈存储的引用。默认情况下，未指定文件路径，并且禁用自定义 Swift 引用。强烈建议配置这个选项，同时将 Swift 存储后端用于镜像存储，因为它避免存储数据库中的凭证。</p> <p> 注意</p> <p>如果将 swift_store_multi_tenant 设为 True，请不要配置此选项。</p> <p>可能的值：</p> <ul style="list-style-type: none">● 代表 glance-api 节点上的绝对路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none">● swift_store_multi_tenant

配置选项 = 默认值	类型	Description
<p>swift_store_container = glance</p>	<p>字符串值</p>	<p>用于存储多个容器的 images/name 前缀的单一容器名称</p> <p>当单个容器用于存储镜像时，此配置选项将指示 Glance 帐户内用于存储所有镜像的容器。当使用多个容器来存储镜像时，这将是所有容器的名称前缀。可以使用配置选项 swift_store_multiple_containers_seed 来控制单/multiple 容器的使用。</p> <p>使用多个容器时，容器将通过为此配置选项设置的值命名，并将镜像 UUID 的第一个 N chars 指定为以下划线分隔的后缀（其中 N 由 swift_store_multiple_containers_seed 指定）。</p> <p>示例：如果 seed 设为 3，swift_store_container = glance，则带有 UUID fdae39a1-bac5-4238-aba4-69bcc726e848 的一个镜像将被放入容器 glance_fda。在创建容器名称时包括在 UUID 中的横线不会计算在字符长度的限制中，因此，当 N= 10 时，容器名称可以为 glance_fdae39a1-ba.</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 如果使用单个容器，此配置选项可以是 Glance Swift 帐户中有效 swift 容器名称的任何字符串 ● 如果使用多个容器，此配置选项可以是任何字符串，只要它满足 Swift 强制执行的容器命名规则。swift_store_multiple_containers_seed 的值也应该考虑。 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_multiple_containers_seed ● swift_store_multi_tenant ● swift_store_create_container_on_put

配置选项 = 默认值	类型	Description
swift_store_create_container_on_put = False	布尔值	<p>上传镜像时，创建容器（如果不存在）。</p> <p>在上传镜像时，如果对应的容器不存在，则会创建它，因为此配置选项被设置为 True。默认情况下，不会创建它。这个行为适用于单一和多个容器模式。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • None
swift_store_endpoint = None	字符串值	<p>用于 Swift 后端存储的 URL 端点。</p> <p>提供代表在 Swift 存储中存储的 Glance 镜像的 URL 端点的字符串值。默认情况下，不会设置端点，并使用 auth 返回的存储 URL。使用 swift_store_endpoint 设置端点会覆盖存储 URL，并用于 Glance 镜像存储。</p> <p> 注意</p> <p>URL 应该包括到的路径，但不包括容器。对象的位置通过将容器和对象附加到配置的 URL 来获取。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 代表到 Swift 容器的有效 URL 路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> • None

配置选项 = 默认值	类型	Description
swift_store_endpoint_type = publicURL	字符串值	<p>Swift 服务的端点类型。</p> <p>此字符串值表示用于获取 Swift 端点的端点类型。端点类型决定了用户被允许执行的操作，例如，读取和写入存储。只有在 swift_store_auth_version 大于 1 时才会使用此设置。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● publicURL ● adminURL ● internalURL <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_endpoint
swift_store_expire_soon_interval = 60	整数值	<p>定义在当前令牌到期前可以请求新令牌的窗口大小的时间（以秒为单位）。</p> <p>通常，Swift 存储驱动程序会在当前令牌过期时获取新令牌，以确保继续访问 Swift。但是，如果令牌实时过期，一些 Swift 事务（如上传镜像段）可能无法正常恢复。</p> <p>因此，通过在当前令牌过期前获取新令牌，请确保令牌不会在尝试事务前过期或接近到期。默认情况下，Swift 存储驱动程序请求新令牌 60 秒或少于当前令牌过期前的请求。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 零 ● 正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_key = None	字符串值	<p>用于针对 Swift 身份验证服务进行身份验证的验证密钥。</p>

配置选项 = 默认值	类型	Description
swift_store_large_object_chunk_size = 200	整数值	<p>镜像数据时段的最大大小（以 MB 为单位）。</p> <p>当镜像数据被分割为上传大于 Swift 集群实施的限制的镜像时，镜像数据被分成不同的片段，它们的大小不大于该配置选项指定的大小。如需了解更多详细信息，请参阅 swift_store_large_object_size。</p> <p>例如：如果 swift_store_large_object_size 为 5GB，而 swift_store_large_object_chunk_size 为 1GB，则大小为 6.2GB 的镜像将被分为 7 段，前 6 个段的大小为 1GB，第七个段的大小为 0.2GB。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数，它小于或等于 Swift 集群强制的大型对象限制。 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_large_object_size
swift_store_large_object_size = 5120	整数值	<p>大小阈值（以 MB 为单位）之后，Glance 将开始分段镜像数据。</p> <p>Swift 对单个上传对象的大小具有上限。默认情况下，这是 5GB。要上传大于这个限制的对象，对象被分成多个与清单文件绑定的较小的对象。如需更多详情，请参阅 https://docs.openstack.org/swift/latest/overview_large_objects.html</p> <p>此配置选项指定 Swift 驱动程序开始将镜像数据段到多个更小文件的大小阈值。目前，Swift 驱动程序只支持创建动态大对象。</p> <div data-bbox="815 1487 922 1621" style="display: inline-block; vertical-align: middle;">  </div> <p style="margin-left: 20px;">注意</p> <p style="margin-left: 20px;">这应该通过考虑 Swift 集群强制的大型对象限制来设置。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数，它小于或等于 Swift 集群强制的大型对象限制。 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_large_object_chunk_size

配置选项 = 默认值	类型	Description
swift_store_multi_tenant = False	布尔值	<p>将镜像存储在租户的 Swift 帐户中。</p> <p>这将启用多租户存储模式，使 Glance 镜像存储在租户特定 Swift 帐户中。如果禁用，Glance 会将所有镜像存储在自己的帐户中。有关多租户存储的详情，请参考 https://wiki.openstack.org/wiki/GlanceSwiftTenantSpecificStorage</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>如果使用多租户 swift 存储，请确保不要使用 <code>swift_store_config_file</code> 选项设置 swift 配置文件。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>swift_store_config_file</code>

配置选项 = 默认值	类型	Description
swift_store_multiple_containers_seed = 0	整数值	<p>指出用于存储镜像的容器数量。</p> <p>在使用单租户存储时，镜像可以存储在一个或多个容器中。当设置为 0 时，所有镜像将存储在一个容器中。当设置为 1 到 32 之间的整数值时，将使用多个容器来存储镜像。此配置选项将确定创建多少个容器。使用的容器总数等于 16^N，因此如果该配置选项设为 2，则使用 $16^2=256$ 容器来存储镜像。</p> <p>有关命名规则的详情，请参阅 swift_store_container。有关使用多个容器的详情，请参考 https://specs.openstack.org/openstack/glance-specs/specs/kilo/swift-store-multiple-containers.html</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>这仅在禁用 <code>swift_store_multi_tenant</code> 时使用。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 小于或等于 32 的非负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_container ● swift_store_multi_tenant ● swift_store_create_container_on_put

配置选项 = 默认值	类型	Description
swift_store_region = None	字符串值	<p>Glance 使用的 Swift 端点区域。</p> <p>提供代表 Swift 区域的字符串值，其中 Glance 可以连接到该区域进行镜像存储。默认情况下，没有设置区域。</p> <p>当 Glance 使用 Swift 作为存储后端存储具有多个端点的特定租户的镜像时，使用 swift_store_region 设置 Swift 区域将允许 Glance 连接到指定区域中的 Swift，而不是单一区域连接。</p> <p>可以为单租户和多租户存储配置此选项。</p> <div data-bbox="815 674 922 840" style="float: left; margin-right: 10px;"> </div> <p>注意</p> <p>使用 swift_store_region 设置区域是特定于租户的，只有在 租户在不同区域上有多个端点时才需要。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表有效 Swift 区域的字符串值。 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_retry_get_count = 0	整数值	<p>在请求失败前，将重试 Swift 下载的次数。</p> <p>提供一个整数值，代表镜像下载的次数，然后才能出错。默认值为 0（在失败的镜像下载时不重试）。当设置为正整数值时，swift_store_retry_get_count 确保在发送错误消息前下载失败时尝试多次下载。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 零 ● 正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
swift_store_service_type = object-store	字符串值	<p>要使用的 Swift 服务类型。</p> <p>提供代表在使用 Swift 后端存储期间存储镜像的服务类型的字符串值。默认服务类型设置为 object-store。</p> <div data-bbox="815 405 922 629" style="float: left; margin-right: 10px;">  </div> <p>注意</p> <p>如果 swift_store_auth_version 设为 2，则此配置选项的值需要是 object-store。如果使用较新版本的 Keystone 或不同的身份验证方案，则可以修改此选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表 Swift 存储的有效服务类型的字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_ssl_compression = True	布尔值	<p>HTTPS Swift 请求的 SSL 层压缩。</p> <p>提供布尔值，以确定是否压缩 SSL 层上镜像的 HTTPS Swift 请求。默认情况下启用压缩。</p> <p>当使用 Swift 作为 Glance 镜像存储的后端存储时，可以使用此选项设置 HTTPS Swift 请求的 SSL 层压缩。如果设置为 False，则禁用 HTTPS Swift 请求的 SSL 层压缩。禁用这个选项可能会提高已压缩格式的镜像的性能，如 qcow2。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
swift_store_use_trusts = True	布尔值	<p>对多租户 Swift 存储使用 trusts。</p> <p>此选项指示 Swift 存储在使用多租户存储时为每个 add/get 请求创建一个信任。使用信任允许 Swift 存储避免在上传或下载数据期间由身份验证令牌过期的问题。</p> <p>默认情况下，swift_store_use_trusts 被设置为 True（启用使用信任）。如果设置为 False，则用户令牌用于 Swift 连接，从而消除了创建信任的开销。</p> <p> 注意</p> <p>只有在 swift_store_multi_tenant 被设置为 True 时，才会考虑此选项</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • swift_store_multi_tenant
swift_store_user = None	字符串值	<p>对 Swift 身份验证服务进行身份验证的用户。</p>
swift_upload_buffer_dir = None	字符串值	<p>上传到 Swift 前用于缓冲镜像片段的目录。</p> <p>提供一个字符串值，代表 glance 节点上目录的绝对路径，在镜像片段上传到 swift 之前会短暂缓冲镜像段。</p> <p>备注：</p> <ul style="list-style-type: none"> • 这只有在配置选项 swift_buffer_on_upload 设置为 True 时才需要。 • 应置备此目录，请记住 swift_store_large_object_chunk_size 以及给定 glance 节点可以同时上传的最大镜像数。 <p>可能的值：</p> <ul style="list-style-type: none"> • 代表绝对路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> • swift_buffer_on_upload • swift_store_large_object_chunk_size

配置选项 = 默认值	类型	Description
vmware_api_retry_count = 10	整数值	<p>VMware API 重试次数。</p> <p>此配置选项指定连接相关问题或服务器 API 调用过载时必须重试 VMware ESX/VC 服务器 API 的次数。无法指定 <i>retry forever</i>。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> None
vmware_ca_file = None	字符串值	<p>CA 捆绑包文件的绝对路径。</p> <p>此配置选项可让操作员使用自定义 Certificate 授权机构文件来验证 ESX/vCenter 证书。</p> <p>如果设置了这个选项，则忽略 "vmware_insecure" 选项，指定的 CA 文件将用于验证 ESX/vCenter 服务器证书，并建立与服务器的安全连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何是 CA 文件的有效绝对路径的字符串 <p>相关选项：</p> <ul style="list-style-type: none"> vmware_insecure

配置选项 = 默认值	类型	Description
vmware_datastores = None	多值	<p>用于存储镜像的数据存储。</p> <p>此配置选项指定镜像存储在 VMWare 存储后端中的数据存储。这个选项可多次指定，以指定多个数据存储。数据存储名称应在其数据中心路径后指定，用 ":" 分隔。可以在数据存储名称后赋予可选权重，再次通过 ":" 分隔来指定优先级。因此，所需格式将变为 <datacenter_path>:<datastore_name>:<optional_weight>。</p> <p>在添加镜像时，会选择权重最高的数据存储，除非已知镜像大小时没有足够的可用空间。如果未指定权重，则假定为 0，目录将被视为最后选择。如果多个数据存储具有相同的权重，则会选择一个带有最多可用空间的权重。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何格式的字符串：<datacenter_path>:<datastore_name>:<optional_weight> <p>相关选项：* None</p>
vmware_insecure = False	布尔值	<p>设置 ESX/vCenter 服务器证书验证。</p> <p>此配置选项使用布尔值来确定是否验证 ESX/vCenter 服务器证书。如果此选项设为 True，则不会验证 ESX/vCenter 服务器证书。如果此选项设置为 False，则使用默认 CA 信任存储进行验证。</p> <p>如果设置了 "vmware_ca_file" 选项，则忽略这个选项。在这种情况下，将使用使用 "vmware_ca_file" 选项指定的文件来验证 ESX/vCenter 服务器证书。</p> <p>可能的值：</p> <ul style="list-style-type: none"> True False <p>相关选项：</p> <ul style="list-style-type: none"> vmware_ca_file

配置选项 = 默认值	类型	Description
vmware_server_host = None	主机地址值	<p>ESX/ESXi 或 vCenter Server 目标系统的地址。</p> <p>此配置选项设置 ESX/ESXi 或 vCenter Server 目标系统的地址。使用 VMware 存储后端时需要这个选项。地址可以包含 IP 地址(127.0.0.1)或 DNS 名称 (www.my-domain.com)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的 IPv4 或 IPv6 地址 ● 有效的 DNS 名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● vmware_server_username ● vmware_server_password
vmware_server_password = None	字符串值	<p>服务器密码。</p> <p>此配置选项使用密码通过 VMware ESX/ESXi 或 vCenter 服务器进行身份验证。使用 VMware 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何字符串，与使用 "vmware_server_username" 选项指定的用户名对应的密码 <p>相关选项：</p> <ul style="list-style-type: none"> ● vmware_server_host ● vmware_server_username
vmware_server_username = None	字符串值	<p>服务器用户名。</p> <p>此配置选项使用用户名来通过 VMware ESX/ESXi 或 vCenter 服务器进行身份验证。使用 VMware 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何字符串，这是具有适当权限的用户的用户名 <p>相关选项：</p> <ul style="list-style-type: none"> ● vmware_server_host ● vmware_server_password

配置选项 = 默认值	类型	Description
vmware_store_image_dir = /openstack_glance	字符串值	<p>glance 镜像要存储在数据存储中的目录。</p> <p>此配置选项指定 glance 镜像存储在 VMware 数据存储中的目录的路径。如果没有设置这个选项，则存储 glance 镜像的默认目录为 openstack_glance。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何是到目录的有效路径的字符串 <p>相关选项：</p> <ul style="list-style-type: none"> None
vmware_task_poll_interval = 5	整数值	<p>用于轮询在 VMware ESX/VC 服务器上调用的远程任务的间隔（以秒为单位）。</p> <p>此配置选项需要休眠时间（以秒为单位）轮询 on-going async 任务，作为 VMWare ESX/VC 服务器 API 调用的一部分。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> None

3.1.12. image_format

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[image_format]` 组下可用的选项。

表 3.11. image_format

配置选项 = 默认值	类型	Description
container_formats = ['ami', 'ari', 'aki', 'bare', 'ovf', 'ova', 'docker', 'compressed']	列表值	<i>container_format</i> image 属性支持的值
disk_formats = ['ami', 'ari', 'aki', 'vhd', 'vhdx', 'vmdk', 'raw', 'qcow2', 'vdi', 'iso', 'ploop']	列表值	<i>disk_format</i> image 属性支持的值

配置选项 = 默认值	类型	Description
vmdk_allowed_types = ['streamOptimized', 'monolithicSparse']	列表值	描述允许的 VMDK <i>创建类型</i> 子格式的字符串列表。建议仅包含 single-file-with-sparse-header 变体，以避免因为处理命名扩展而造成潜在的主机文件暴露。如果此列表为空，则不允许使用 VMDK 镜像类型。请注意，这目前仅在镜像转换期间检查（如果已启用），并限制我们将从其转换的 VMDK 镜像的类型。

3.1.13. keystone_authtoken

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[keystone_authtoken]` 组下可用的选项。

表 3.12. keystone_authtoken

配置选项 = 默认值	类型	Description
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
auth_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。 这个选项已弃用，现在使用 <code>www_authenticate_uri</code>，并将在 S 发行版本中删除。弃用自：Queens *reason:* auth_uri 选项已弃用，现在使用 <code>www_authenticate_uri</code>，并将在 S 发行版本中删除。
auth_version = None	字符串值	Identity API 端点的 API 版本。
cache = None	字符串值	存储 Swift 缓存对象的请求环境密钥。当使用 Swift 缓存部署 <code>auth_token</code> 中间件时，请使用此选项使中间件共享具有 <code>swift</code> 的缓存后端。否则，使用 <code>memcached_servers</code> 选项。
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。默认为系统 CA。
certFile = None	字符串值	如果身份服务器需要客户端证书，则需要此项
delay_auth_decision = False	布尔值	不要在中间件内处理授权请求，而是将授权决策委派给下游 WSGI 组件。

配置选项 = 默认值	类型	Description
enforce_token_bind = permissive	字符串值	用于控制令牌绑定的使用和类型。可以设置为 "disabled" 以不检查令牌绑定。如果绑定类型是已知服务器的绑定类型，则验证绑定信息，并忽略它（如 "permissive"），但如果绑定类型未知，则令牌将被拒绝。"必需"令牌绑定需要允许任何形式的令牌绑定。最后，令牌中必须存在的绑定方法名称。
http_connect_timeout = None	整数值	请求超时值与身份 API 服务器通信。
http_request_max_retries = 3	整数值	与身份 API 服务器通信时，我们尝试重新连接的次数。
include_service_catalog = True	布尔值	（可选）指示是否设置 X-Service-Catalog 标头。如果为 False，则中间件不会要求在令牌验证时请求服务目录，且不会设置 X-Service-Catalog 标头。
insecure = False	布尔值	验证 HTTPS 连接。
interface = internal	字符串值	用于 Identity API 端点的接口。有效值为 "public", "internal"（默认）或 "admin"。
keyfile = None	字符串值	如果身份服务器需要客户端证书，则需要此项
memcache_pool_connection_timeout = 10	整数值	（可选）操作等待从池中获取 memcached 客户端连接的秒数。
memcache_pool_dead_retry = 300	整数值	（可选）在再次尝试前 memcached 服务器被视为 dead 的秒数。
memcache_pool_maxsize = 10	整数值	（可选）到每个 memcached 服务器的最大打开连接数。
memcache_pool_socket_timeout = 3	整数值	（可选）与 memcached 服务器通信的套接字超时（以秒为单位）。
memcache_pool_unused_timeout = 60	整数值	（可选）与 memcached 的连接在池关闭前未使用的秒数。
memcache_secret_key = None	字符串值	（可选，如果定义了 memcache_security_strategy）此字符串用于密钥派生。
memcache_security_strategy = None	字符串值	（可选）如果定义，则指示令牌数据是否应进行身份验证或加密。如果 MAC，令牌数据在缓存中进行身份验证（使用 HMAC）。如果 ENCRYPT，令牌数据会在缓存中加密并进行身份验证。如果该值不是这些选项之一或为空，则 auth_token 会在初始化时引发异常。

配置选项 = 默认值	类型	Description
memcache_use_advanced_pool = False	布尔值	(可选) 使用 advanced (事件安全) memcached 客户端池。高级池只能在 python 2.x 下工作。
memcached_servers = None	列表值	(可选) 指定要用于缓存的 memcached 服务器列表。如果未定义, 令牌将会被缓存在进程中。
region_name = None	字符串值	找到身份服务器的区域。
service_token_roles = ['service']	列表值	需要出现在服务令牌中的角色。服务令牌可以请求可以使用过期的令牌, 因此此检查应严格控制只有实际服务应发送此令牌。此处的角色作为 ANY 检查应用, 因此此列表中的任何角色都必须存在。为了向后兼容的原因, 这目前只会影响 allow_expired 检查。
service_token_roles_required = False	布尔值	为了向后兼容的原因, 我们必须允许有效的服务令牌传递, 但不允许将 service_token_roles 检查视为有效。在以后的发行版本中设置此 true 将成为默认设置, 并应启用 (如果可能)。
service_type = None	字符串值	服务的名称或类型 (在服务目录中出现)。这用于验证具有受限访问规则的令牌。
token_cache_time = 300	整数	为了防止过度努力验证令牌, 中间件会缓存之前可预见的令牌 (以秒为单位)。设置为 -1 可完全禁用缓存。
www_authenticate_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点, 因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下, 这个端点应该被未指定版本, 但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点, 则该服务用户不应该用于验证令牌的端点, 因为普通用户可能无法访问该端点。

3.1.14. oslo_concurrency

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[oslo_concurrency]` 组下可用的选项。

表 3.13. oslo_concurrency

配置选项 = 默认值	类型	Description
disable_process_locking = False	布尔值	启用或禁用进程间锁定。

配置选项 = 默认值	类型	Description
lock_path = None	字符串值	用于锁定文件的目录。为安全起见，指定目录应只由运行需要锁定的进程的用户写入。默认为环境变量 OSLO_LOCK_PATH。如果使用外部锁定，则必须设置锁定路径。

3.1.15. oslo_messaging_amqp

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[oslo_messaging_amqp]` 组下可用的选项。

表 3.14. oslo_messaging_amqp

配置选项 = 默认值	类型	Description
addressing_mode = dynamic	字符串值	表示驱动程序使用的寻址模式。允许的值： <i>legacy</i> - 使用旧的不可路由寻址； <i>routable</i> - 使用可路由地址； <i>dynamic</i> - 如果消息总线不支持路由，使用 <i>legacy</i> ，如果支持，使用 <i>routable</i> 。
anycast_address = anycast	字符串值	在发送到一组消费者时附加到地址前缀。消息总线用来识别应在消费者之间以轮循方式发送的消息。
broadcast_prefix = broadcast	字符串值	广播到所有服务器时使用的地址前缀
connection_retry_backoff = 2	整数值	每次失败故障转移尝试后，将 <code>connection_retry_interval</code> 增加到这秒数。
connection_retry_interval = 1	整数值	尝试重新连接前暂停的秒数。
connection_retry_interval_max = 30	整数值	<code>connection_retry_interval</code> + <code>connection_retry_backoff</code> 的最大限制
container_name = None	字符串值	AMQP 容器的名称必须全局唯一。默认为生成的 UUID
default_notification_exchange = None	字符串值	通知地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_notification_exchange if set else control_exchange if set else <i>notify</i>
default_notify_timeout = 30	整数值	发送通知消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_reply_retry = 0	整数值	重新发送回复消息的最大尝试次数，因为可恢复错误而失败。

配置选项 = 默认值	类型	Description
default_reply_timeout = 30	整数值	rpc 回复消息发送的截止时间。
default_rpc_exchange = None	字符串值	RPC 地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_rpc_exchange if set other control_exchange if set else <i>rpc</i>
default_send_timeout = 30	整数值	rpc cast 或调用消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_sender_link_timeout = 600	整数值	调度闲置发送者链接清除的持续时间。过期后分离链接。
group_request_prefix = unicast	字符串值	发送到组中任何服务器时的地址前缀
idle_timeout = 0	整数值	不活跃连接的超时时间（以秒为单位）
link_retry_delay = 10	整数值	重新连接 AMQP 1.0 链接之间的暂停时间，该链接因为可恢复的错误而失败。
multicast_address = multicast	字符串值	在发送异常消息时，附加到地址前缀。消息总线用来识别 fanout 消息。
notify_address_prefix = openstack.org/om/notify	字符串值	所有生成的通知地址的地址前缀
notify_server_credit = 100	整数值	传入通知消息的窗口大小
pre_settled = ['rpc-cast', 'rpc-reply']	多值	发送此类型的消息预先组装。预先设置的消息不会接收来自 peer 的确认信息。请注意：如果发送失败，则预先设置的消息可能会被静默丢弃。允许的值： <i>rpc-call</i> - 发送 RPC Calls <i>pre-settled</i> <i>rpc-reply</i> - 发送 RPC Replies <i>pre-settled</i> <i>rpc-cast</i> - Send RPC Casts <i>pre-settled</i> <i>notify</i> - Send Notifications <i>pre-settled</i>
pseudo_vhost = True	布尔值	为不支持虚拟主机的消息总线启用虚拟主机支持（如 <i>qpidd</i> ）。当设置为 <i>true</i> 时，虚拟主机名将添加到所有消息总线地址中，从而有效地为每个虚拟主机 <i>创建专用子网</i> 。如果消息总线支持使用 AMQP 1.0 Open performative 中的 <i>hostname</i> 字段作为虚拟主机的名称，设置为 <i>False</i> 。
reply_link_credit = 200	整数值	传入 RPC 消息的窗口大小。

配置选项 = 默认值	类型	Description
rpc_address_prefix = openstack.org/om/rpc	字符串值	所有生成的 RPC 地址的地址前缀
rpc_server_credit = 100	整数值	传入 RPC Request 消息的窗口大小
<code>`sasl_config_dir = `</code>	字符串值	包含 SASL 配置的目录路径
<code>`sasl_config_name = `</code>	字符串值	配置文件名称（不带 .conf 后缀）
<code>`sasl_default_realm = `</code>	字符串值	如果用户名中没有 realm，则要使用的 SASL 域
<code>`sasl_mechanisms = `</code>	字符串值	可接受的 SASL 机制的以空格分隔的列表
server_request_prefix = exclusive	字符串值	发送到特定服务器时使用的地址前缀
ssl = False	布尔值	尝试通过 SSL 连接。如果没有给出其他与 ssl 相关的参数，它将使用系统的 CA-bundle 来验证服务器证书。
<code>`ssl_ca_file = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_cert_file = `</code>	字符串值	自我标识用于客户端身份验证的证书 PEM 文件
<code>`ssl_key_file = `</code>	字符串值	用于为 ssl_cert_file 证书签名的私钥 PEM 文件（可选）
ssl_key_password = None	字符串值	用于解密 ssl_key_file 的密码（如果加密）
ssl_verify_vhost = False	布尔值	默认情况下，SSL 检查服务器证书中的名称与 transport_url 中的主机名匹配。在某些配置中，最好使用虚拟主机名。例如，如果服务器使用 Server Name Indication TLS extension (rfc6066) 来为每个虚拟主机提供证书。如果服务器的 SSL 证书使用虚拟主机名称而不是 DNS 名称，则将 ssl_verify_vhost 设置为 True。
trace = False	布尔值	debug：将 AMQP 帧转储到 stdout
unicast_address = unicast	字符串值	在发送到特定 RPC/Notification 服务器时，附加到地址前缀。消息总线用来识别发送到单个目的地的消息。

3.1.16. oslo_messaging_kafka

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[oslo_messaging_kafka]` 组下可用的选项。

表 3.15. oslo_messaging_kafka

配置选项 = 默认值	类型	Description
compression_codec = none	字符串值	producer 生成的所有数据的压缩 codec。如果没有设置，则不会使用压缩。请注意，这允许的值取决于 kafka 版本
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
consumer_group = oslo_messaging_consumer	字符串值	Kafka 消费者的组 ID。一个组中的消费者将协调消息消耗
enable_auto_commit = False	布尔值	启用异步消费者提交
kafka_consumer_timeout = 1.0	浮点值	Kafka 用户的默认超时
kafka_max_fetch_bytes = 1048576	整数值	Kafka consumer 的最大获取字节数
max_poll_records = 500	整数值	轮询调用中返回的最大记录数
pool_size = 10	整数值	Kafka Consumers 的池大小
producer_batch_size = 16384	整数值	producer async 发送的批处理大小
producer_batch_timeout = 0.0	浮点值	以秒为单位的 KafkaProducer 批处理延迟上限
sasl_mechanism = PLAIN	字符串值	当安全协议是 SASL 时的机制
security_protocol = PLAINTEXT	字符串值	用于与代理通信的协议
<code>`ssl_cafile = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_client_cert_file = `</code>	字符串值	用于身份验证的客户端证书 PEM 文件。
<code>`ssl_client_key_file = `</code>	字符串值	用于身份验证的客户端密钥 PEM 文件。
<code>`ssl_client_key_password = `</code>	字符串值	用于身份验证的客户端密钥密码文件。

3.1.17. oslo_messaging_notifications

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[oslo_messaging_notifications]` 组下可用的选项。

表 3.16. oslo_messaging_notifications

配置选项 = 默认值	类型	Description
<code>driver = []</code>	多值	处理发送通知的驱动程序。可能的值有 <code>messaging</code> , <code>messagingv2</code> , <code>routing</code> , <code>log</code> , <code>test</code> , <code>noop</code>
<code>retry = -1</code>	整数值	重新发送因为可恢复错误而发送失败的通知消息的最大尝试次数。0 - 不重试, -1 - indefinite
<code>topics = ['notifications']</code>	列表值	用于 OpenStack 通知的 AMQP 主题。
<code>transport_url = None</code>	字符串值	代表用于通知的消息传递驱动程序的 URL。如果没有设置, 我们回退到用于 RPC 的同一配置。

3.1.18. oslo_messaging_rabbit

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[oslo_messaging_rabbit]` 组下可用的选项。

表 3.17. oslo_messaging_rabbit

配置选项 = 默认值	类型	Description
<code>amqp_auto_delete = False</code>	布尔值	AMQP 中的自动删除队列。
<code>amqp_durable_queues = False</code>	布尔值	在 AMQP 中使用持久队列。
<code>direct_mandatory_flag = True</code>	布尔值	(DEPRECATED)为直接发送启用/禁用 RabbitMQ 强制标记。直接发送用作回复, 因此在客户端队列不存在时引发 <code>MessageUndeliverable</code> 异常。 <code>MessageUndeliverable</code> 异常将用于循环使发送者恢复。此标志已弃用, 且无法停用此功能
<code>enable_cancel_on_failover = False</code>	布尔值	启用 <code>x-cancel-on-ha-failover</code> 标志, 使 <code>rabbitmq</code> 服务器将取消并通知 <code>consumerswhen</code> 队列停机
<code>heartbeat_in_pthread = False</code>	布尔值	默认情况下, 通过原生 <code>python</code> 线程运行健康检查心跳线程。如果此选项等于 <code>False</code> , 则健康检查 <code>heartbeat</code> 将继承父进程的执行模型。例如, 如果父进程使用 <code>eventlet/greenlet</code> 对 <code>stdlib</code> 进行补丁, 则心跳将通过绿色线程运行。这个选项应该只针对 <code>wsgi</code> 服务设置为 <code>True</code> 。
<code>heartbeat_rate = 2</code>	整数值	<code>heartbeat_timeout_threshold</code> 期间的我们检查心跳的频率。

配置选项 = 默认值	类型	Description
heartbeat_timeout_thresh old = 60	整数值	如果 heartbeat 的 keep-alive 失败(0 禁用 heartbeat), 则 Rabbit 代理被视为 down 的秒数。
kombu_compression = None	字符串值	EXPERIMENTAL : Possible 值为 : gzip、bz2。如果没有设置压缩, 则不会使用。这个选项可能在以后的版本中不可用。
kombu_failover_strategy = round-robin	字符串值	确定当当前连接到的 RabbitMQ 节点不可用时如何选择下一个 RabbitMQ 节点。仅在配置中提供多个 RabbitMQ 节点时生效。
kombu_missing_consumer_retry_timeout = 60	整数值	在放弃发送其回复前等待缺少的客户端的时间。这个值不应比 rpc_response_timeout 长。
kombu_reconnect_delay = 1.0	浮点值	响应 AMQP 消费者取消通知前等待多久。
rabbit_ha_queues = False	布尔值	尝试在 RabbitMQ 中使用 HA 队列(x-ha-policy: all)。如果更改这个选项, 您必须擦除 RabbitMQ 数据库。在 RabbitMQ 3.0 中, 在声明队列时, x-ha-policy 参数不再控制队列镜像。如果您只想确保所有队列(带有自动生成的名称的队列除外)都在所有节点上进行镜像, 请运行: "rabbitmqctl set_policy HA ^(?!amq\.) gem {"ha-mode": "all"}"
rabbit_interval_max = 30	整数值	RabbitMQ 连接重试的最大间隔。默认值为 30 秒。
rabbit_login_method = AMQPLAIN	字符串值	RabbitMQ 登录方法。
rabbit_qos_prefetch_count = 0	整数值	指定要预抓取的消息数量。设置为零可允许无限消息。
rabbit_retry_backoff = 2	整数值	连接到 RabbitMQ 时重试之间的 backoff 时长。
rabbit_retry_interval = 1	整数值	使用 RabbitMQ 重试连接的频率。
rabbit_transient_queues_ttl = 1800	整数值	正整数代表队列 TTL (x-expires)的持续时间(以秒为单位)。TTL 期间未使用的队列将被自动删除。参数仅影响 reply 和 fanout 队列。
ssl = False	布尔值	通过 SSL 连接。
<code>`ssl_ca_file = `</code>	字符串值	SSL 证书颁发机构文件(仅在启用了 SSL 时评估)。
<code>`ssl_cert_file = `</code>	字符串值	SSL 证书文件(仅在启用了 SSL 时评估)。

配置选项 = 默认值	类型	Description
<code>`ssl_key_file = `</code>	字符串值	SSL 密钥文件（仅在启用了 SSL 时 valid）。
<code>`ssl_version = `</code>	字符串值	要使用的 SSL 版本（仅在启用了 SSL 时评估）。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1_1 和 TLSv1_2 可能在某些发行版中提供。

3.1.19. oslo_middleware

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[oslo_middleware]` 组下可用的选项。

表 3.18. oslo_middleware

配置选项 = 默认值	类型	Description
<code>enable_proxy_headers_parsing = False</code>	布尔值	应用程序是否在代理后面。这决定了中间件是否应该解析标头。

3.1.20. oslo_policy

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[oslo_policy]` 组下可用的选项。

表 3.19. oslo_policy

配置选项 = 默认值	类型	描述
<code>enforce_new_defaults = False</code>	布尔值	这个选项控制在评估策略时是否使用旧的弃用默认值。如果为 True ，则不会评估旧的已弃用的默认值。这意味着，如果旧默认值允许任何现有令牌，但不允许使用新默认值，它将被禁止。我们鼓励启用此标志以及 <code>enforce_scope</code> 标志，以便您可以同时获得新默认值和 <code>scope_type</code> 的好处。
<code>enforce_scope = False</code>	布尔值	此选项控制在评估策略时是否强制执行范围。如果为 True ，则请求中使用的令牌范围将与被强制执行策略的 <code>scope_types</code> 进行比较。如果范围不匹配，则引发 <code>InvalidScope</code> 异常。如果为 False ，则会记录一条信息，告知操作员使用了不匹配的范围调用策略。
<code>policy_default_rule = default</code>	字符串值	默认规则。未找到请求的规则时强制执行。
<code>policy_dirs = ['policy.d']</code>	多值	存储策略配置文件的目录。它们可以相对于 <code>config_dir</code> 选项或绝对路径定义的搜索路径中的任何目录。由 <code>policy_file</code> 定义的文件必须存在，才能搜索这些目录。缺少或空目录将被忽略。
<code>policy_file = policy.yaml</code>	字符串值	将角色映射到给定服务权限的文件的相对或绝对路径。必须使用与配置文件设置相关的相对路径。

配置选项 = 默认值	类型	描述
remote_content_type = application/x-www-form-urlencoded	字符串值	用于发送和接收基于 REST 策略检查的数据的内容类型
remote_ssl_ca_cert_file = None	字符串值	基于 REST 的策略检查的 ca cert 文件的绝对路径
remote_ssl_client_cert_file = None	字符串值	基于 REST 的策略检查的客户端证书的绝对路径
remote_ssl_client_key_file = None	字符串值	绝对路径客户端密钥文件 REST 基于策略检查
remote_ssl_verify_server_cert = False	布尔值	基于 REST 的策略检查的服务器身份验证

3.1.21. paste_deploy

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[paste_deploy]` 组下可用的选项。

表 3.20. paste_deploy

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
config_file = None	字符串值	<p>粘贴配置文件的名称。</p> <p>提供代表粘贴配置文件名称的字符串值，以用于为服务器应用程序部署配置 pipelines。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 为粘贴配置文件提供与 glance 目录相关的名称或路径，而不是绝对路径。 ● 大多数情形中不需要编辑 Glance 附带的示例粘贴配置文件，因为它附带所有常见部署类别的现成管道。 <p>如果没有为此选项指定值，则会在已知配置目录中搜索含有相应 Glance 服务配置文件名称前缀的 paste.ini 文件。（例如，如果缺少此选项或 glance-api.conf 中没有设置值，该服务将查找名为 glance-api-paste.ini 的文件。）如果没有找到粘贴配置文件，服务将不会启动。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表粘贴配置文件名称的字符串值。 <p>相关选项：</p> <ul style="list-style-type: none"> ● flavor
flavor = None	字符串值	<p>服务器应用管道中使用的部署类别。</p> <p>提供代表服务器应用程序 pipeline 中使用的适当部署类别的字符串值。这通常是粘贴配置文件中管道的部分名称，删除了服务名称。</p> <p>例如，如果您的粘贴配置文件中的部分名称为 [pipeline:glance-api-keystone]，请将 flavor 设置为 keystone。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表部分管道名称的字符串值。 <p>相关选项：</p> <ul style="list-style-type: none"> ● config_file

3.1.22. profiler

下表概述了 `/etc/glance/glance-api.conf` 文件中的 **[profiler]** 组下可用的选项。

表 3.21. profiler

配置选项 = 默认值	类型	Description
connection_string = messaging://	字符串值	<p>通知程序后端的连接字符串。</p> <p>默认值为 messaging://，它将通知程序设置为 <code>oslo_messaging</code>。</p> <p>可能的值示例：</p> <ul style="list-style-type: none"> ● messaging:// - 使用 <code>oslo_messaging</code> 驱动程序来发送范围。 ● redis://127.0.0.1:6379 - 使用 <code>redis</code> 驱动程序发送 <code>span</code>。 ● MongoDB://127.0.0.1:27017 - 使用 <code>mongodb</code> 驱动程序进行发送范围。 ● Elasticsearch://127.0.0.1:9200 - 使用 <code>elasticsearch</code> 驱动程序来发送 <code>span</code>。 ● Jaeger://127.0.0.1:6831 - 使用 <code>jaeger tracing</code> 作为发送 <code>span</code> 的驱动程序。
enabled = False	布尔值	<p>为此节点上的所有服务启用性能分析。</p> <p>默认值为 <code>False</code>（完全禁用性能分析功能）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● <code>true</code>：启用该功能 ● <code>false</code>：禁用该功能。分析无法通过此项目操作启动。如果性能分析由另一个项目触发，则此项目部分将为空。
es_doc_type = notification	字符串值	elasticsearch 中通知索引的文档类型。
es_scroll_size = 10000	整数值	Elasticsearch 批量分割大型请求。此参数定义每个批处理的最大大小（例如： <code>es_scroll_size=10000</code> ）。
es_scroll_time = 2m	字符串值	此参数是一个时间值参数（例如： <code>es_scroll_time=2m</code> ），指示参与搜索的节点将维护相关资源以继续和支持它。

配置选项 = 默认值	类型	Description
filter_error_trace = False	布尔值	<p>启用包含 error/exception 的过滤器追踪来分隔的位置。</p> <p>默认值为 False。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用包含 error/exception 的过滤器追踪。 ● false：禁用过滤器。
hmac_keys = SECRET_KEY	字符串值	<p>用于加密上下文数据的 secret 密钥，以用于性能分析。</p> <p>此字符串值应该具有以下格式：<key1>[,<key2>,...<keyn>]，其中每个键都是随机字符串。通过 REST API 触发性能分析的用户必须在 REST API 调用的标头中设置其中一个键，使其包含此节点的性能分析结果。</p> <p>"enabled"标志和"hmac_keys"配置选项都应设置为启用性能分析。此外，为了跨所有服务生成正确的性能分析信息，OpenStack 项目之间至少需要一个关键。这样可确保它可用于客户端生成 trace，其中包含所有可能的资源的信息。</p>
sentinel_service_name = mymaster	字符串值	<p>Redis sentinel 使用服务名称来识别 master redis 服务。此参数定义名称（例如：sentinal_service_name=mymaster）。</p>
socket_timeout = 0.1	浮点值	<p>Redis sentinel 在连接上提供了一个 timeout 选项。此参数定义该超时（例如：socket_timeout=0.1）。</p>
trace_sqlalchemy = False	布尔值	<p>在服务中启用 SQL 请求性能分析。</p> <p>默认值为 False (SQL 请求不会被跟踪)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用 SQL 请求性能分析。每个 SQL 查询都将是 trace 的一部分，可以通过花费的时间来分析。 ● false：禁用 SQL 请求性能分析。花费的时间仅显示在更高级别的操作上。无法以这种方式分析单个 SQL 查询。

3.1.23. store_type_location_strategy

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[store_type_location_strategy]` 组下可用的选项。

表 3.22. store_type_location_strategy

配置选项 = 默认值	类型	Description
store_type_preference = []	列表值	<p>存储后端的首选顺序。</p> <p>按照应从存储后端检索镜像的顺序提供以逗号分隔的存储名称列表。这些存储名称必须使用 stores 配置选项注册。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>只有在将 store_type 选择为 location_strategy 配置选项的值时，才会应用 store_type_preference 配置选项。空列表不会更改位置顺序。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 空列表 ● 以逗号分隔的注册存储名称列表。法律值为： <ul style="list-style-type: none"> ○ file ○ http ○ rbd ○ swift ○ cinder ○ vmware <p>相关选项：</p> <ul style="list-style-type: none"> ● location_strategy ● 存储

3.1.24. task

下表概述了 `/etc/glance/glance-api.conf` 文件中的 **[task]** 组下可用的选项。

表 3.23. task

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
task_executor = taskflow	字符串值	<p>用于运行任务脚本的任务 executor。</p> <p>提供代表任务执行的 executor 的字符串值。默认情况下使用 TaskFlow executor。</p> <p>TaskFlow 帮助使任务执行简单、一致、可扩展且可靠。它还允许创建轻量级任务对象和/或功能，它们以声明性方式合并到流中。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● taskflow <p>相关选项：</p> <ul style="list-style-type: none"> ● None
task_time_to_live = 48	整数值	<p>任务在成功或失败的时间（以小时为单位）</p>
work_dir = None	字符串值	<p>用于异步任务操作的工作目录的绝对路径。</p> <p>这里设置的目录将用于对镜像进行操作 - 通常在目标存储中导入之前。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>为 work_dir 提供值时，请确保为并发任务提供足够的空间，以便在不用尽空间的情况下运行并发任务。</p> </div> </div> <p>粗略估算可通过使用平均镜像大小（如 500MB）的 max_workers 数量相乘。镜像大小估算应该根据您的部署的平均大小进行。请注意，根据运行的任务，您可能需要根据任务所执行的操作，按一些因素来乘以这个数字。例如，如果启用了镜像转换，您可能希望将可用大小加倍。所有这点是说的，请记住，只需要估算，您应该根据最糟糕的情况进行这些操作，并在出现问题时做好准备。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表工作目录的绝对路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

3.1.25. taskflow_executor

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[taskflow_executor]` 组下可用的选项。

表 3.24. taskflow_executor

配置选项 = 默认值	类型	Description
conversion_format = None	字符串值	<p>设置所需的镜像转换格式。</p> <p>提供您希望在存储 Glance 之前要转换镜像的有效镜像格式，供 Glance 使用。对于特定的存储后端，需要适当的镜像格式转换，以便有效地处理存储基础架构的带宽和使用。</p> <p>默认情况下，不会设置 conversion_format，且必须在配置文件中明确设置。</p> <p>此选项允许的值有 raw、qcow2 和 vmdk。raw 格式是无结构的磁盘格式，应在 RBD 或 Ceph 存储后端用于镜像存储时选择。QEMU 模拟器支持 qcow2，它动态扩展并支持写时复制。vmdk 是许多常见虚拟机监视器（如 VMWare 工作站）支持的另一种常见磁盘格式。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● qcow2 ● raw ● vmdk <p>相关选项：</p> <ul style="list-style-type: none"> ● disk_formats
engine_mode = parallel	字符串值	<p>设置 taskflow 引擎模式。</p> <p>提供字符串 type 值，以设置任务流引擎将任务调度到主机上的 worker 的模式。基于此模式，引擎可以在单一或多个线程中执行任务。此配置选项的可能值为：serial 和 parallel。当设置为 serial 时，引擎在单个线程中运行所有任务，从而按顺序执行任务。把它设置为 并行 使引擎在多个线程中运行任务。这会引导并行执行任务。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● serial ● parallel <p>相关选项：</p> <ul style="list-style-type: none"> ● max_workers

配置选项 = 默认值	类型	Description
max_workers = 10	整数值	<p>设置引擎可执行任务的数量。</p> <p>提供整数值来限制主机上可实例化的 worker 数量。换句话说，这个数字定义了任务流引擎可以同时执行的并行任务数量。当引擎模式设置为并行时，这个值可以大于一。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 大于或等于 1 的整数值 <p>相关选项：</p> <ul style="list-style-type: none"> engine_mode

3.1.26. wsgi

下表概述了 `/etc/glance/glance-api.conf` 文件中的 `[wsgi]` 组下可用的选项。

表 3.25. wsgi


配置选项 = 默认值	类型	描述
python_interpreter = /usr/bin/python3	字符串值	生成外部进程时要使用的 python 解释器路径。默认情况下，这是 <code>sys.executable</code> ，它应当与运行 Glance 本身的解释程序相同。然而，在某些情况下（如 <code>uwsgi</code> ）实际上并不指向 python 解释器本身。
task_pool_threads = 16	整数值	池中处理异步任务的线程数量（每个 worker 进程）。这控制了每个 worker 一次可以运行的异步任务（例如，镜像可互操作性导入）。如果这太大，则每个 worker 和/或您可能还增加内存占用量，例如磁盘或出站网络带宽等其他系统资源。 如果这太小，镜像导入请求必须等待线程可用才能开始处理。

3.2. GLANCE-SCRUBBER.CONF

本节包含 `/etc/glance/glance-scrubber.conf` 文件的选项。

3.2.1. DEFAULT

下表概述了 `/etc/glance/glance-scrubber.conf` 文件中的 `[DEFAULT]` 组下可用的选项。

配置选项 = 默认值	类型	Description
allow_additional_image_properties = True	布尔值	<p>允许用户为镜像添加额外的/自定义属性。</p> <p>Glance 定义在每个镜像上显示的标准属性集合（在其架构中）。这些属性也称为基本属性。除了这些属性，Glance 允许用户向镜像添加自定义属性。它们被称为附加属性。</p> <p>默认情况下，此配置选项被设置为 True，允许用户添加额外的属性。可以通过 image_property_quota 配置选项控制可添加到镜像中的额外属性数量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • image_property_quota <p>弃用自：U ssuri</p> <p>reason: 这个选项是多余的。通过 <i>image_property_quota</i> 配置选项控制自定义镜像属性的使用。这个选项计划在 Victoria 开发周期中删除。</p>
api_limit_max = 1000	整数值	<p>请求可以返回的最大结果数。</p> <p>如 limit_param_default 的帮助文本中所述，一些请求可能会返回多个结果。要返回的结果数量取决于请求中的 limit 参数或 limit_param_default 配置选项。两个情况下的值都不能大于该配置选项定义的绝对最大值。任何大于这个值的都会修剪到此处定义的最大值。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>把它设置为非常大的值可能会减慢数据库查询并增加响应时间。把它设置为非常低的值可能会导致用户体验不佳。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> • 任何正整数 <p>相关选项：</p> <ul style="list-style-type: none"> • limit_param_default

配置选项 = 默认值	类型	Description
daemon = False	布尔值	<p>作为守护进程运行 scrubber。</p> <p>此布尔值配置选项指示 scrubber 是否应该作为长时间唤醒的长时间运行来清理镜像。可使用配置选项 wakeup_time 指定唤醒间隔。</p> <p>如果此选项设置为 False（默认值），则清理操作会运行一次来清理镜像并退出。在这种情况下，如果操作员希望实施镜像持续清理，则需要将清理网络调度为 cron 作业。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • wakeup_time
debug = False	布尔值	<p>如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。</p>

配置选项 = 默认值	类型	Description
<pre> default_log_levels = ['amqp=WARN', 'amqp-lib=WARN', 'boto=WARN', 'qp-id=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']</pre>	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 log_config_append，则忽略此选项。

配置选项 = 默认值	类型	Description
delayed_delete = False	布尔值	<p>打开/关闭延迟删除。</p> <p>通常，当镜像被删除时，glance-api 服务会将镜像置于 已删除 状态，并同时删除其数据。延迟删除是 Glance 中的一个功能，它可以延迟删除镜像数据直到稍后的时间点（由配置选项 scrub_time 决定）。当打开延迟删除时，glance-api 服务会将镜像置于 pending_delete 状态，并将镜像数据保留在存储后端中，以便镜像清理网络在以后被删除。镜像清理网络将在成功删除镜像数据时将镜像移到 已删除 的状态。</p> <div data-bbox="815 651 922 815" style="display: inline-block; vertical-align: top;">  </div> <p style="margin-left: 20px;">注意</p> <p style="margin-left: 20px;">当打开延迟删除时，镜像清理网络必须作为定期任务运行，以防止后端存储使用不必要的使用量填满。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● scrub_time ● wakeup_time ● scrub_pool_size

配置选项 = 默认值	类型	Description
digest_algorithm = sha256	字符串值	<p>用于数字签名的摘要算法。</p> <p>提供代表用于生成数字签名的摘要算法的字符串值。默认情况下使用 sha256。</p> <p>要获得您的平台上 OpenSSL 版本支持的可用算法列表，请运行以下命令：openssl list-message-digest-algorithms。示例是 <i>sha1</i>、<i>sha256</i> 和 <i>sha512</i>。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>digest_algorithm 与 Glance 的镜像签名和验证无关。它仅用于签署通用唯一标识符(UUID)，作为证书文件和密钥文件验证的一部分。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● OpenSSL 消息摘要算法标识符 <p>关系选项：</p> <ul style="list-style-type: none"> ● None
enabled_import_methods = ['glance-direct', 'web-download', 'copy-image']	列表值	<p>启用的镜像导入方法列表</p> <p>'glance-direct', 'copy-image' and 'web-download' are enabled by default.</p> <p>Related options: ** [DEFAULT]/node_staging_uri</p>

配置选项 = 默认值	类型	Description
enforce_secure_rbac = False	布尔值	<p>根据 OpenStack 中使用的常见用户角色定义，实施 API 访问。启用此选项规范化特定于项目的读/写操作，如创建私有镜像或更新 member 角色后共享镜像的状态。它还规范了一个可用于特定于项目的 API 操作的只读变体，如在 reader 角色后列出项目中的私有镜像。</p> <p>Operator 应该有机会了解 glance 的新镜像策略，在其部署中审核分配，并使用 keystone 中的默认角色（如 admin、member 和 reader）更新权限。</p> <p>相关选项：</p> <ul style="list-style-type: none"> • [oslo_policy]/enforce_new_defaults <p>弃用自：Wallaby</p> <p>原因：已经引进了此选项，要求操作员根据常见的 RBAC 用户角色选择强制授权，这是 Wallaby 发行版本的 EXPERIMENTAL。此行为将是未来的版本中的默认设置和 STABLE，允许删除这个选项。</p>
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。

配置选项 = 默认值	类型	Description
hashing_algorithm = sha512	字符串值	<p>用于计算 <i>os_hash_value</i> 属性的安全哈希算法。</p> <p>此选项配置 Glance "multihash"，它由两个镜像属性组成：<i>os_hash_algo</i> 和 <i>os_hash_value</i>。<i>os_hash_algo</i> 将由此配置选项的值填充，当算法应用到上传或导入的镜像数据时，<i>os_hash_value</i> 将由 <i>hexdigest</i> 填充。</p> <p>该值必须是 python <i>hashlib</i> 库识别的有效安全哈希算法名称。您可以通过检查 Glance 安装中使用的库的 <i>hashlib.algorithms_available</i> 数据成员来确定这些内容。然而，为了实现互操作性，我们建议您使用由 <i>hashlib.algorithms_guaranteed</i> 数据成员提供的安全散列名称集，因为所有平台上的 <i>hashlib</i> 库都可以保证支持这些算法。因此，在本地使用 <i>hashlib</i> 的任何镜像消费者都应能够验证镜像的 <i>os_hash_value</i>。</p> <p><i>sha512</i> 的默认值是一个性能安全哈希算法。</p> <p>如果此选项配置不正确，任何尝试存储镜像数据都将失败。因此，我们建议使用默认值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● Python <i>hashlib</i> 库识别的任何安全哈希算法名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
image_location_quota = 10	整数值	<p>镜像上允许的最大位置数。</p> <p>任何负值都解释为无限。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● None
image_member_quota = 128	整数值	<p>每个镜像的最大镜像成员数量。</p> <p>这限制了镜像可以被共享的最大用户。任何负值都解释为无限。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
image_property_quota = 128	整数值	<p>镜像上允许的最大属性数。</p> <p>这对镜像可以具有的额外属性数量强制实施上限。任何负值都解释为无限。</p> <div style="display: flex; align-items: flex-start;">  <div> <p>注意</p> <p>如果禁用附加属性，则这不会影响。请参阅 allow_additional_image_properties。</p> </div> </div> <p>相关选项：</p> <ul style="list-style-type: none"> ● allow_additional_image_properties
image_size_cap = 1099511627776	整数值	<p>用户可以上传的最大镜像大小（以字节为单位）。</p> <p>大于此处提到的大小的镜像上传会导致镜像创建失败。此配置选项默认为 1099511627776 字节(1 TiB)。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 这个值只有在仔细考虑后才会增加，且必须设置小于或等于 8 EiB (9223372036854775808)。 ● 必须谨慎考虑后端存储容量设置这个值。把它设置为非常低的值可能会导致大量镜像失败。然后，将其设置为一个非常大的值可能会导致存储更快地消耗。因此，这必须根据创建的镜像的性质和可用的存储容量来设置。 <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何小于或等于 9223372036854775808 的正数
image_tag_quota = 128	整数值	<p>镜像上允许的最大标签数。</p> <p>任何负值都解释为无限。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● None
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。

配置选项 = 默认值	类型	Description
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。
limit_param_default = 25	整数值	<p>为请求返回的默认结果数。</p> <p>对特定 API 请求的响应（如列出镜像）可能会返回多个项目。通过在 API 请求中指定 limit 参数，可以显式控制返回的结果数量。但是，如果没有指定 limit 参数，则此配置值将用作任何 API 请求返回的默认结果数。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 此配置选项的值可能不大于 api_limit_max 指定的值。 ● 把它设置为非常大的值可能会减慢数据库查询并增加响应时间。把它设置为非常低的值可能会导致用户体验不佳。 <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>api_limit_max</code>
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 <code>log-date-format</code> ）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。

配置选项 = 默认值	类型	Description
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 DEBUG 时，要附加到日志消息的其他数据。 oslo_log.formatters.ContextFormatter 使用
logging_default_format_s tring = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_for mat = %(user) s %(tenant) s %(domain) s % (user_domain) s % (project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。

配置选项 = 默认值	类型	Description
metadata_encryption_key = None	字符串值	<p>用于加密存储位置元数据的 AES 密钥。</p> <p>提供代表用于加密 Glance 存储元数据的 AES 密码的字符串值。</p>  <p>注意</p> <p>要使用的 AES 密钥必须设置为随机字符串，长度为 16、24 或 32 字节。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 代表有效 AES 密钥的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> None
node_staging_uri = file:///tmp/staging/	字符串值	<p>URL 提供临时存储临时数据的位置</p> <p>此选项仅用于 Glance 内部用途。Glance 将在镜像导入过程中保存用户上传的镜像数据到 <i>暂存</i> 端点。</p> <p>这个选项不会以任何方式更改 <i>staging</i> API 端点。</p>  <p>注意</p> <p>不建议使用与 [task]/work_dir 相同的路径</p>  <p>注意</p> <p><code>file://<absolute-directory-path></code> 是现在唯一支持的 <code>api_image_import</code> 流。</p>  <p>注意</p> <p>暂存路径必须位于所有 Glance API 节点可用的共享文件系统上。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 以 <code>file://</code> 开始的字符串，后跟绝对 FS 路径 <p>相关选项：</p> <ul style="list-style-type: none"> [task]/work_dir
publish_errors = False	布尔值	启用或禁用错误事件的发布。

配置选项 = 默认值	类型	Description
pydev_worker_debug_host = None	主机地址值	<p>pydev 服务器的主机地址。</p> <p>提供代表用于调试的 pydev 服务器的主机名或 IP 的字符串值。pydev 服务器侦听此地址上的调试连接，促进 Glance 中的远程调试。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效主机名 ● 有效的 IP 地址 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
pydev_worker_debug_port = 5678	端口值	<p>pydev 服务器将侦听的端口号。</p> <p>提供将 pydev 服务器绑定到的端口号。pydev 进程接受此端口上的调试连接，并有助于在 Glance 中进行远程调试。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效端口号 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
restore = None	字符串值	<p>将镜像状态从 <i>pending_delete</i> 恢复到 <i>active</i>。</p> <p>当镜像被错误删除且 Glance 中启用了 <i>pending delete</i> 功能，管理员可利用此选项将镜像的状态从 <i>pending_delete</i> 重置为 <i>active</i>。请确保在恢复镜像前停止 glance-scrubber 守护进程，以避免镜像数据不一致。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 镜像的 uuid

配置选项 = 默认值	类型	Description
scrub_pool_size = 1	整数值	<p>用于清理镜像的线程池的大小。</p> <p>当大量镜像可用于清理时，可以并行清理镜像，以便清理队列保持控制，后端存储会及时回收。此配置选项表示并行清理的最大镜像数。默认值为 one，表示串行清理。上面的任何值都表示并行清理。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何非零正整数 <p>相关选项：</p> <ul style="list-style-type: none"> delayed_delete
scrub_time = 0	整数值	<p>延迟镜像清理的时间（以秒为单位）。</p> <p>当延迟删除被打开时，镜像会在删除时放入 pending_delete 状态，直到清理时间删除其镜像数据。通常，在镜像进入 pending_delete 状态后，就可以进行清理。但是，清理可能会延迟，直到稍后使用这个配置选项为止。此选项表示镜像在可用于清理前处于 pending_delete 状态的时间周期。</p> <p>务必要意识到，这对存储有影响。scrub_time 越大，从已删除镜像回收后端存储所需的时间越长。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何非负整数 <p>相关选项：</p> <ul style="list-style-type: none"> delayed_delete

配置选项 = 默认值	类型	Description
<p>show_image_direct_url = False</p>	<p>布尔值</p>	<p>在返回镜像时显示直接镜像位置。</p> <p>此配置选项指示在向用户返回镜像详情时是否显示直接镜像位置。直接镜像位置是镜像数据存储在后端存储中的位置。此镜像位置显示在镜像属性 direct_url 下。</p> <p>当镜像存在多个镜像位置时，会根据配置选项 location_strategy 指示的位置策略显示最佳位置。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 显示镜像位置可能会显示 GRAVE SECURITY RISK，因为镜像位置有时可以包含凭证。因此，默认设置为 False。使用 EXTREME CAUTION 把它设置为 True，仅 IF 您知道自己正在做什么！ ● 如果运算符希望避免向用户显示任何镜像位置，则此选项和 show_multiple_locations MUST 都设为 False。 <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● show_multiple_locations ● location_strategy

配置选项 = 默认值	类型	Description
show_multiple_locations = False	布尔值	<p>显示返回镜像时的所有镜像位置。</p> <p>此配置选项指示在向用户返回镜像详情时是否显示所有镜像位置。当镜像存在多个镜像位置时，根据配置 opt location_strategy 指示的位置策略进行排序。镜像位置显示在镜像属性 位置 下。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 显示镜像位置可能会显示 GRAVE SECURITY RISK，因为镜像位置有时可以包含凭证。因此，默认设置为 False。使用 EXTREME CAUTION 把它设置为 True，仅 IF 您知道自己正在做什么！ ● 如需更多信息，请参阅 https://wiki.openstack.org/wiki/OSSN/OSSN-0065。 ● 如果运算符希望避免向用户显示任何镜像位置，则此选项和 show_image_direct_url MUST 都被设置为 False。 <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● show_image_direct_url ● location_strategy <p>弃用自：Newton</p> <p>*reason:*Use of option（自 Newton 起已弃用）是一个安全风险，当我们找出满足目前需要它的用例后，将删除它。之前宣布可使用策略以更高的粒度实现相同的功能不正确。您无法在当前时间通过策略配置临时解决此选项，但这是我们认为该修复将采取的方向。请留意 Glance 发行注记，以保持在这个问题的最新进度。</p>
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。

配置选项 = 默认值	类型	Description
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 <code>log_config_append</code> ，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 <code>log_config_append</code> ，则忽略此选项。
user_storage_quota = 0	字符串值	<p>每个租户的最大镜像存储量。</p> <p>这会强制对所有存储中的所有租户镜像使用的累积存储进行上限。这是每个租户的限制。</p> <p>此配置选项的默认单位是字节。但是，可以使用区分大小写的字面 B、KB、MB、GB 和 TB 指定存储单元，分别代表 Bytes、KiloBytes、MegaBytes、GigaBytes 和 TeraBytes。请注意，值和单元之间不应有任何空格。值 0 表示没有配额强制。负值无效，并导致错误。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 一个字符串，它是一个非负整数的有效串联，代表存储值，以及代表上面提到的存储单元的可选字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> None

配置选项 = 默认值	类型	Description
wakeup_time = 300	整数值	<p>scrubber 在守护进程模式下运行之间的时间间隔（以秒为单位）。</p> <p>scrubber 可以作为 cron 作业或守护进程运行。当作为守护进程运行时，此配置时间指定了两个运行之间的时间段。当唤醒清理程序时，它会考虑 scrub_time，获取并清理所有可以被清理的 pending_delete 镜像。</p> <p>如果唤醒时间时间设置为一个非常大的值，则每次运行时可能会清理大量镜像。另外，这会影响到后端存储的重新声明速度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何非负整数 <p>相关选项：</p> <ul style="list-style-type: none"> daemon delayed_delete
watch-log-file = False	布尔值	<p>使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 <code>log_file</code> 选项并使用 Linux 平台时，才有意义。如果设置了 <code>log_config_append</code>，则忽略此选项。</p>
worker_self_reference_url = None	字符串值	<p>此 worker 的 URL。</p> <p>如果设置了此设置，其他 glance 工作程序将了解如何直接联系此 worker。对于镜像导入，单个 worker 分阶段镜像和其他 worker 需要能够将导入请求代理到正确的请求。</p> <p>如果未设置，这被视为 public_endpoint，这通常会设置为所有 worker 上的相同值，从而有效地禁用代理行为。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 可以从其他 worker 访问此 worker 的 URL <p>相关选项：</p> <ul style="list-style-type: none"> <code>public_endpoint</code>

3.2.2. database

下表概述了 `/etc/glance/glance-scrubber.conf` 文件中的 **[database]** 组下可用的选项。

表 3.26. database

配置选项 = 默认值	类型	Description
backend = sqlalchemy	字符串值	用于数据库的后端。
connection = None	字符串值	用于连接到数据库的 SQLAlchemy 连接字符串。
connection_debug = 0	整数值	SQL 调试信息的详细程度：0=None, 100=Everything。
<code>connection_parameters = `</code>	字符串值	用于连接时附加到连接 URL 的可选 URL 参数；指定 param1=value1¶m2=value2&...
connection_recycle_time = 3600	整数值	连接池中存在的连接的时间超过这个秒数，下一次从池中签出它们的时间将替换为新的连接。
connection_trace = False	布尔值	将 Python 堆栈跟踪作为注释字符串添加到 SQL。
db_inc_retry_interval = True	布尔值	如果为 True，将数据库操作的重试间隔增加到 db_max_retry_interval。
db_max_retries = 20	整数值	在引发错误前出现连接错误或死锁错误时的最大重试次数。设置为 -1 以指定无限重试计数。
db_max_retry_interval = 10	整数值	如果设置了 db_inc_retry_interval，则数据库操作重试之间的最大秒数。
db_retry_interval = 1	整数值	数据库事务重试之间的秒数。
max_overflow = 50	整数值	如果设置，请将这个值用于 SQLAlchemy 的 max_overflow。
max_pool_size = 5	整数值	在池中保留的最大 SQL 连接数。将值设为 0 表示没有限制。
max_retries = 10	整数值	启动期间数据库连接重试的最大数量。设置为 -1 以指定无限重试计数。
mysql_enable_ndb = False	布尔值	如果为 True，则透明地启用对处理 MySQL 集群 (NDB) 的支持。
mysql_sql_mode = TRADITIONAL	字符串值	用于 MySQL 会话的 SQL 模式。这个选项（包括默认选项）会覆盖任何 server-set SQL 模式。要使用服务器配置设置的任何 SQL 模式，请将其设置为 no 值。示例：mysql_sql_mode=
pool_timeout = None	整数值	如果设置，请将这个值用于 SQLAlchemy 的 pool_timeout。

配置选项 = 默认值	类型	Description
retry_interval = 10	整数值	重新尝试打开 SQL 连接之间的间隔。
slave_connection = None	字符串值	用于连接到从数据库的 SQLAlchemy 连接字符串。
sqlite_synchronous = True	布尔值	如果为 True，则 SQLite 使用同步模式。
use_db_reconnect = False	布尔值	启用在连接丢失时进行数据库重新连接的实验性使用。
use_tpool = False	布尔值	启用所有 DB API 调用的线程池的实验性使用

3.2.3. glance_store

下表概述了 `/etc/glance/glance-scrubber.conf` 文件中的 `[glance_store]` 组下可用的选项。

表 3.27. glance_store

配置选项 = 默认值	类型	Description
cinder_api_insecure = False	布尔值	<p>允许对 cinder 执行不安全的 SSL 请求。</p> <p>如果此选项被设置为 True，则使用 cinder_ca_certificates_file 选项指定的 CA 证书文件验证 HTTPS 端点连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • cinder_ca_certificates_file
cinder_ca_certificates_file = None	字符串值	<p>用于 Cinder 客户端请求的 CA 证书文件的位置。</p> <p>指定的 CA 证书文件（如果设置）用于通过 HTTPS 端点验证 Cinder 连接。如果端点是 HTTP，则忽略这个值。cinder_api_insecure 必须设置为 True 以启用验证。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • ca 证书文件的路径 <p>相关选项：</p> <ul style="list-style-type: none"> • cinder_api_insecure

配置选项 = 默认值	类型	Description
cinder_catalog_info = volume3::publicURL	字符串值	<p>在服务目录中查找 cinder 时匹配的信息。</p> <p>如果没有设置 cinder_endpoint_template，并且任何 cinder_store_auth_address，cinder_store_user_name，cinder_store_project_name，cinder_store_password 没有设置，cinder 存储将使用此信息从当前上下文中的服务目录中查找 Cinder 端点。如果设置，将考虑 cinder_os_region_name 来获取适当的端点。</p> <p>服务目录可以通过 openstack catalog list 命令列出。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 应指定以下形式的字符串：< service_type>:<service_name>:<interface> At least service_type 和 interface。 service_name 可以省略。 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_os_region_name ● cinder_endpoint_template ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password

配置选项 = 默认值	类型	Description
cinder_endpoint_template = None	字符串值	<p>使用 cinder 端点的模板覆盖服务目录查找。</p> <p>设置此选项时，这个值用于生成 cinder 端点，而不是从服务目录查找。如果指定了 cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, 和 cinder_store_password, 则忽略这个值。</p> <p>如果设置了此配置选项，则忽略 cinder_catalog_info。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● cinder 端点的 URL 模板字符串，其中 %%(tenant) 替换为当前的租户（项目）名称。例如： http://cinder.openstack.example.org/v2/%%(tenant)s <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password ● cinder_catalog_info
cinder_enforce_multipath = False	布尔值	<p>如果设置为 True，则当 multipathd 未运行时，镜像传输的卷附加将中止。否则，它将回退到单一路径。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True 或 False <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_use_multipath
cinder_http_retries = 3	整数值	<p>失败 http 调用的 cinderclient 重试次数。</p> <p>当调用被任何错误失败时，cinderclient 会在睡眠几秒钟后重试调用。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
cinder_mount_point_base = /var/lib/glance/mnt	字符串值	<p>NFS 卷挂载在 glance 节点上的目录。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表挂载点的绝对路径的字符串。
cinder_os_region_name = None	字符串值	<p>从服务目录查找 cinder 服务的区域名称。</p> <p>这仅在使用 cinder_catalog_info 来确定端点时使用。如果设置，则此节点的 cinder 端点的查找将过滤到指定的区域。当目录中列出了多个区域时，它很有用。如果没有设置，则会从每个区域查找端点。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 是有效区域名称的字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_catalog_info
cinder_state_transition_timeout = 300	整数值	<p>等待 cinder 卷转换完成的时间（以秒为单位）。</p> <p>当创建、删除或附加到 glance 节点以读/写卷数据时，卷的状态将改变。例如，新创建的卷状态会在 创建 过程完成后变为 available。这将指定等待状态更改的最长时间。如果在等待时发生超时，或者状态更改为非预期的值（如 错误），则镜像创建会失败。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
cinder_store_auth_address = None	字符串值	<p>cinder 身份验证服务正在侦听的地址。</p> <p>当指定所有 cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, 和 cinder_store_password 选项时, 指定的值始终用于身份验证。这有助于通过将镜像服务存储在特定于镜像服务的项目/租户中, 从用户隐藏镜像卷。它还允许用户在 glance 的 ACL 控制下与其他项目共享镜像卷。</p> <p>如果没有设置任何一个选项, 则 cinder 端点会从服务目录中查找, 并使用当前上下文的用户和项目。</p> <p>可能的值 :</p> <ul style="list-style-type: none"> 有效的身份验证服务地址, 例如 : http://openstack.example.org/identity/v2.0 <p>相关选项 :</p> <ul style="list-style-type: none"> cinder_store_user_name cinder_store_password cinder_store_project_name
cinder_store_password = None	字符串值	<p>用于针对 Cinder 进行身份验证的用户密码。</p> <p>这必须与以下所有相关选项一起使用。如果未指定其中任何一个, 则使用当前上下文的用户。</p> <p>可能的值 :</p> <ul style="list-style-type: none"> 由 cinder_store_user_name 指定的用户的有效密码 <p>相关选项 :</p> <ul style="list-style-type: none"> cinder_store_auth_address cinder_store_user_name cinder_store_project_name

配置选项 = 默认值	类型	Description
cinder_store_project_name = None	字符串值	<p>镜像卷的项目名称存储在 cinder 中。</p> <p>如果没有设置此配置选项，则使用当前上下文中的项目。</p> <p>这必须与以下所有相关选项一起使用。如果未指定其中任何一个，则使用当前上下文的项目。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效项目名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_password
cinder_store_user_name = None	字符串值	<p>要针对 cinder 进行身份验证的用户名。</p> <p>这必须与以下所有相关选项一起使用。如果未指定其中任何一个，则使用当前上下文的用户。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的用户名 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_password ● cinder_store_project_name
cinder_use_multipath = False	布尔值	<p>在部署中支持或不支持识别 mutipath 的标记。</p> <p>如果不支持多路径，则将其设置为 False。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True 或 False <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_enforce_multipath

配置选项 = 默认值	类型	Description
<p>cinder_volume_type = None</p>	<p>字符串值</p>	<p>在 cinder 中用于创建卷的卷类型。</p> <p>有些 Cinder 后端可以有几个卷类型来优化存储使用。添加此选项可让操作器在 cinder 中选择特定卷类型，该类型可以针对镜像进行了优化。</p> <p>如果没有设置，则 cinder 配置中指定的默认卷类型将用于创建卷。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 来自 cinder 的有效卷类型 <p>相关选项：</p> <ul style="list-style-type: none"> ● None <div data-bbox="815 792 922 1048" style="border: 1px solid gray; padding: 5px; width: fit-content;">  </div> <p>注意</p> <p>您不能使用与 NFS 后端关联的加密 volume_type。当 glance_store 试图写入或访问存储在该卷中的镜像数据时，存储在 NFS 后端上的加密卷将引发异常。请参考 Cinder 管理员确定适当的 volume_type。</p>

配置选项 = 默认值	类型	Description
<p>default_store = file</p>	<p>字符串值</p>	<p>用于存储镜像的默认方案。</p> <p>提供代表用于存储镜像的默认方案的字符串值。如果没有设置，Glance 使用 file 作为默认方案来使用文件存储镜像。</p> <div data-bbox="815 436 922 600" style="float: left; margin-right: 10px;">  </div> <p>注意</p> <p>为这个配置选项提供的值必须是使用 stores 配置选项注册的存储的有效方案。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● file ● filesystem ● http ● https ● swift ● swift+http ● swift+https ● swift+config ● rbd ● cinder ● vsphere ● s3 <p>相关选项：</p> <ul style="list-style-type: none"> ● 存储 <p>弃用自：Rocky</p> <p>reason: 这个选项根据新配置选项 "default_backend" 弃用，它的行为与 "default_store" 配置选项类似。</p> <p>计划在 U 开发周期中移除此选项。</p>

配置选项 = 默认值	类型	Description
default_swift_reference = ref1	字符串值	<p>引用默认 Swift 帐户/备份存储参数。</p> <p>提供代表使用 swift 帐户/反馈存储所需默认参数集合的引用的字符串值。此配置选项的默认引用值为 <i>ref1</i>。此配置选项解引用参数，并有助于每次添加新镜像时 Swift 存储后端中的镜像存储。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
filesystem_store_chunk_size = 65536	整数值	<p>块大小，以字节为单位。</p> <p>读取或写入镜像文件时使用的块大小。增加这个值可以提高吞吐量，但在处理大量请求时，它也可能稍微增加内存用量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
filesystem_store_datadir = /var/lib/glance/images	字符串值	<p>文件系统后端存储写入镜像的目录。</p> <p>启动时，Glance 会创建该目录（如果不存在），并验证在其下运行 glance-api 的用户的写入访问权限。如果写入访问不可用，则引发 BadStoreConfiguration 异常，并且文件系统存储可能无法添加新镜像。</p> <div data-bbox="815 510 922 824" style="border: 1px solid gray; padding: 5px; width: fit-content;">  </div> <p>注意</p> <p>此目录仅在将文件系统存储用作存储后端时使用。filesystem_store_datadir 或 filesystem_store_datadirs 选项必须在 glance-api.conf 中指定。如果指定了这两个选项，则会引发 BadStoreConfiguration，并且文件系统存储可能无法添加新镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 到目录的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● filesystem_store_datadirs ● filesystem_store_file_perm

配置选项 = 默认值	类型	Description
filesystem_store_datadirs = None	多值	<p>文件系统后端存储写入镜像的目录及其优先级列表。</p> <p>文件系统存储可以配置为将镜像存储在多个目录中，而不是使用由 filesystem_store_datadir 配置选项指定的单个目录。使用多个目录时，可以为每个目录赋予一个可选优先级，以指定应使用它们的首选项顺序。priority 是一个整数，它将与目录路径连接，冒号越高，代表优先级更高。当两个目录具有相同的优先级时，会使用具有最多可用空间的目录。如果没有指定优先级，则默认为零。</p> <p>有关使用多个存储目录配置文件系统存储的更多信息，请参阅 https://docs.openstack.org/glance/latest/configuration/configuring.html</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 40px; height: 40px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, #ccc 2px, #ccc 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>此目录仅在将文件系统存储用作存储后端时使用。filesystem_store_datadir 或 filesystem_store_datadirs 选项必须在 glance-api.conf 中指定。如果指定了这两个选项，则会引发 BadStoreConfiguration，并且文件系统存储可能无法添加新镜像。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 以下格式的字符串列表： <ul style="list-style-type: none"> ○ <a valid directory path>:<optional integer priority> <p>相关选项：</p> <ul style="list-style-type: none"> ● filesystem_store_datadir ● filesystem_store_file_perm

配置选项 = 默认值	类型	Description
filesystem_store_file_permission = 0	整数值	<p>镜像文件的文件访问权限。</p> <p>为镜像数据设置预期的文件访问权限。这提供了一种方式，可以让其他服务（如 Nova）直接使用文件系统存储中的镜像。运行要被授予访问权限的服务的用户，可以成为拥有创建文件的组的成员。为这个配置选项分配一个小于或等于 0 的值表示没有对默认权限进行任何更改。这个值将被解码为八进制数。</p> <p>如需更多信息，请参阅文档 https://docs.openstack.org/glance/latest/configuration/configuring.html</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的文件访问权限 ● 零 ● 任何负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
filesystem_store_metadata_file = None	字符串值	<p>文件系统存储元数据文件。</p> <p>包含与文件系统存储关联的任何位置返回元数据的文件路径。设置此选项后，它仅用于之后创建的新镜像 - 以前现有镜像不受影响。</p> <p>该文件必须包含有效的 JSON 对象。该对象应包含键 id 和 mountpoint。两个键的值都应该是一个字符串。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 存储元数据文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
filesystem_thin_provisioning = False	布尔值	<p>在这个后端中启用或不精简置备。</p> <p>这个配置选项启用了在文件系统中没有真正写入 null 字节序列的功能，文件系统会自动解释为 null 字节，且不会实际消耗您的存储。启用此功能也会加快镜像上传速度，并节省后端中的空间，因为不会通过网络发送 null 字节序列。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • None
http_proxy_information = {}	dict 值	<p>用于连接到远程服务器的 http/https 代理信息。</p> <p>这个配置选项指定应该用来连接到远程服务器的 http/https 代理信息。代理信息应该是方案和代理的键值对，如 http:10.0.0.1:3128。您还可以通过使用逗号分隔键值对（如 http:10.0.0.1:3128, https:10.0.0.1:1080）来指定多个方案的代理。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 如前文所述，以逗号分隔的 scheme:proxy 对列表 <p>相关选项：</p> <ul style="list-style-type: none"> • None
https_ca_certificates_file = None	字符串值	<p>CA 捆绑包文件的路径。</p> <p>此配置选项可让操作员使用自定义证书颁发机构文件来验证远程服务器证书。如果设置了这个选项，则忽略 https_insecure 选项，指定的 CA 文件将用于验证服务器证书并建立到服务器的安全连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • CA 文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> • https_insecure

配置选项 = 默认值	类型	Description
https_insecure = True	布尔值	<p>设置远程服务器证书的验证。</p> <p>此配置选项使用布尔值来确定是否验证远程服务器证书。如果设置为 True，则不会验证远程服务器证书。如果选项设为 False，则使用默认 CA 信任存储进行验证。</p> <p>如果设置了 https_ca_certificates_file，则忽略这个选项。然后，使用 https_ca_certificates_file 选项指定的文件验证远程服务器证书。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● https_ca_certificates_file
rados_connect_timeout = 0	整数值	<p>连接到 Ceph 集群的超时值。</p> <p>此配置选项在连接到 Ceph 集群时所用的超时值（以秒为单位）。它会设置在关闭连接前等待 glance-api 的时间。这可防止在连接到 RBD 时出现 glance-api 挂起。如果此选项的值设为小于或等于 0，则不设置超时，并且使用默认的 librados 值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
<code>`rbd_store_ceph_conf = `</code>	字符串值	<p>Ceph 配置文件路径。</p> <p>此配置选项指定要使用的 Ceph 配置文件的路径。如果此选项的值未由用户设置或设置为空字符串，librados 将按顺序搜索默认的 Ceph 配置文件位置来读取标准 <code>ceph.conf</code> 文件。详情请参阅 Ceph 文档。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>如果使用 Cephx 身份验证，此文件应该在 <code>client.<USER></code> 部分中包含对正确密钥环的引用</p> </div> </div> <p>注意 2：如果您将此选项留空（默认），使用的实际 Ceph 配置文件可能会根据使用 librados 的版本而有所不同。如果您知道哪个配置文件生效非常重要，您可以在此处使用这个选项指定该文件。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 配置文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>rbd_store_user</code>
<code>rbd_store_chunk_size = 8</code>	整数值	<p>大小（以 MB 为单位）将 RADOS 镜像块到其中。</p> <p>提供代表 Glance 镜像的大小（以 MB 为单位）的整数值。默认块大小为 8MB。为获得最佳性能，该值应该是 2 的指数。</p> <p>当 Ceph 的 RBD 对象存储系统用作存储 Glance 镜像的存储后端时，镜像将使用此选项集中到设置的大小的对象中。然后，这些块的对象存储在用于 Glance 的分布式块数据存储中。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
rbd_store_pool = images	字符串值	<p>存储镜像的 RADOS 池。</p> <p>当 RBD 用作存储 Glance 镜像的存储后端时，镜像通过将对象（镜像块）的逻辑分组存储到池中。每个池使用它可以包含的放置组数量来定义。使用的默认池是 <i>images</i>。</p> <p>有关 RBD 存储后端的更多信息，请参阅： http://ceph.com/planet/how-data-is-stored-in-ceph-cluster/</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效池名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
rbd_store_user = None	字符串值	<p>RADOS 用户以以下身份进行身份验证：</p> <p>此配置选项在 RADOS 用户进行身份验证，以如下方式进行身份验证：这仅在启用 RADOS 身份验证时且仅在用户使用 Cephx 身份验证时才需要。如果此选项的值未由用户设置或设置为 None，则会选择一个默认值，该值将基于 <code>rbd_store_ceph_conf</code> 中的 <code>client.</code> 部分。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的 RADOS 用户 <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>rbd_store_ceph_conf</code>

配置选项 = 默认值	类型	Description
rdp_thin_provisioning = False	布尔值	<p>在这个后端中启用或不精简置备。</p> <p>此配置选项允许在 RBD 后端上实际写入 null 字节序列的功能，Ceph 将自动显示的漏洞被 Ceph 解释为 null 字节，且不会实际消耗您的存储。启用此功能也会加快镜像上传速度，并节省后端中的空间，因为不会通过网络发送 null 字节序列。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • None
rootwrap_config = /etc/glance/rootwrap.conf	字符串值	<p>用于以 root 身份运行命令的 rootwrap 配置文件的路径。</p> <p>cinder 存储需要 root 特权才能运行镜像卷（用于连接 iSCSI/FC 卷并读取/写入卷数据等）。配置文件应允许 cinder 存储和 os-brick 库所需的命令。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • rootwrap 配置文件的路径 <p>相关选项：</p> <ul style="list-style-type: none"> • None
s3_store_access_key = None	字符串值	<p>S3 查询令牌访问密钥。</p> <p>此配置选项使用访问密钥来与 Amazon S3 或 S3 兼容存储服务器进行身份验证。使用 S3 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 任何字符串值，这是具有适当特权的用户的访问密钥 <p>相关选项：</p> <ul style="list-style-type: none"> • s3_store_host • s3_store_secret_key

配置选项 = 默认值	类型	Description
s3_store_bucket = None	字符串值	<p>用于存储 Glance 数据的 S3 存储桶。</p> <p>此配置选项指定 glance 镜像存储在 S3 中的位置。如果 s3_store_create_bucket_on_put 设为 true，它将自动创建，即使存储桶不存在。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_create_bucket_on_put ● s3_store_bucket_url_format
s3_store_bucket_url_format = auto	字符串值	<p>用于确定对象的 S3 调用格式。</p> <p>此配置选项使用访问模型，用于在 S3 存储桶中指定对象地址。</p> <p>注意：在 路径- 样式中，对象的端点类似于 https://s3.amazonaws.com/bucket/example.img。在 虚拟- 样式中，对象的端点类似于 https://bucket.s3.amazonaws.com/example.img。如果您没有在存储桶名称中遵循 DNS 命名惯例，您可以使用路径风格获取对象，但不能以虚拟样式获取对象。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● auto、虚拟 或 path 的任何字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_bucket
s3_store_create_bucket_on_put = False	布尔值	<p>确定 S3 是否应该创建新存储桶。</p> <p>此配置选项使用布尔值来指示 Glance 是否应该向 S3 创建新存储桶（如果不存在）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何布尔值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
s3_store_host = None	字符串值	<p>S3 服务器侦听的主机。</p> <p>此配置选项设置 S3 或 S3 兼容存储服务器的主机。使用 S3 存储后端时需要这个选项。主机可以包含 DNS 名称（如 s3.amazonaws.com、my-object-storage.com）或 IP 地址(127.0.0.1)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的 DNS 名称 ● 有效的 IPv4 地址 <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_access_key ● s3_store_secret_key
s3_store_large_object_chunk_size = 10	整数值	<p>在上传部分时，S3 应该使用什么多部分上传部分大小（以 MB 为单位）。</p> <p>这个配置选项使用镜像分割大小（以 MB 为单位）进行多部分上传。</p> <p>注意：您只能分成 10,000 个镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值（必须大于或等于 5M） <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_large_object_size ● s3_store_thread_pools
s3_store_large_object_size = 100	整数值	<p>S3 启动镜像文件的大小应该以 MB 为单位，并在 S3 中进行多部分上传。</p> <p>此配置选项采用以 MB 为单位的阈值，以确定是否将镜像上传到 S3，还是将其分割（多部分上传）。</p> <p>注意：您只能分成 10,000 个镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_large_object_chunk_size ● s3_store_thread_pools

配置选项 = 默认值	类型	Description
s3_store_secret_key = None	字符串值	<p>S3 查询令牌 secret 密钥。</p> <p>此配置选项使用 secret 密钥来与 Amazon S3 或 S3 兼容存储服务器进行身份验证。使用 S3 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串值，它是与使用 s3_store_host 选项指定的访问密钥对应的 secret 键 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_host s3_store_access_key
s3_store_thread_pools = 10	整数值	<p>在 S3 中执行多部分上传的线程池数量。</p> <p>此配置选项在执行多部分上传时取线程池的数量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_large_object_size s3_store_large_object_chunk_size


配置选项 = 默认值	类型	Description
stores = ['file', 'http']	列表值	<p>启用 Glance 存储的列表。</p> <p>注册存储后端，以逗号分隔列表的形式用于存储磁盘镜像。为使用 Glance 存储磁盘镜像而启用的默认存储是 file 和 http。</p> <p>可能的值：</p> <ul style="list-style-type: none">● 一个以逗号分隔的列表，其中包括：<ul style="list-style-type: none">○ file○ http○ swift○ rbd○ cinder○ vmware○ s3 <p>相关选项：</p> <ul style="list-style-type: none">● default_store <p>弃用自： Rocky</p> <p>reason: 这个选项根据新配置选项 "enabled_backends" 弃用，这有助于配置不同方案的多个后端存储。</p> <p>计划在 U 开发周期中移除此选项。</p>

配置选项 = 默认值	类型	Description
swift_buffer_on_upload = False	布尔值	<p>上传到 Swift 前缓冲区镜像段。</p> <p>提供布尔值，以指示 Glance 是否应该在上传到 swift 时将镜像数据缓冲到磁盘。这可使 Glance 恢复出错时上传。</p> <p>注意：当启用这个选项时，应该非常小心，因为这会增加 API 节点上的磁盘用量。请注意，根据文件系统的配置方式，用于缓冲的磁盘空间可能会减少 glance 镜像缓存可用的实际磁盘空间。磁盘利用率将根据以下等情况设置：</p> <p>(swift_store_large_object_chunk_size * workers * 1000)</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • swift_upload_buffer_dir
swift_store_admin_tenants = []	列表值	<p>将被授予 admin 访问权限的租户列表。</p> <p>这是租户列表，将在多租户模式下授予 Glance 创建的所有 Swift 容器的读取/写入访问权限。默认值为空列表。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 以逗号分隔的字符串列表，代表 Keystone 项目/租户的 UUID <p>相关选项：</p> <ul style="list-style-type: none"> • None
swift_store_auth_address = None	字符串值	Swift 身份验证服务侦听的端口。

配置选项 = 默认值	类型	Description
swift_store_auth_insecure = False	布尔值	<p>设置服务器证书验证。</p> <p>此布尔值决定了是否验证服务器证书。如果此选项设置为 True，则 swiftclient 在验证时不会检查有效的 SSL 证书。如果选项设为 False，则使用默认 CA 信任存储进行验证。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • swift_store_cacert
swift_store_auth_version = 2	字符串值	<p>要使用的身份验证服务的版本。对于 keystone 有效版本是 2 和 3，对于 swauth 和 rackspace 为 1（已弃用）。</p>
swift_store_cacert = None	字符串值	<p>CA 捆绑包文件的路径。</p> <p>此配置选项可让操作员指定自定义证书颁发机构文件的路径，以便在连接到 Swift 时进行 SSL 验证。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • CA 文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> • swift_store_auth_insecure

配置选项 = 默认值	类型	Description
<p>swift_store_config_file = None</p>	<p>字符串值</p>	<p>包含 swift 帐户配置的文件的路径。</p> <p>包括一个字符串值，代表配置文件的路径，该文件对配置的每个 Swift 帐户/反馈存储的引用。默认情况下，未指定文件路径，并且禁用自定义 Swift 引用。强烈建议配置这个选项，同时将 Swift 存储后端用于镜像存储，因为它避免存储数据库中的凭证。</p> <div data-bbox="815 510 922 645" style="float: left; margin-right: 10px;"> </div> <p>注意</p> <p>如果将 swift_store_multi_tenant 设为 True，请不要配置此选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表 glance-api 节点上的绝对路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_multi_tenant

配置选项 = 默认值	类型	Description
<p>swift_store_container = glance</p>	<p>字符串值</p>	<p>用于存储多个容器的 images/name 前缀的单一容器名称</p> <p>当单个容器用于存储镜像时，此配置选项将指示 Glance 帐户内用于存储所有镜像的容器。当使用多个容器来存储镜像时，这将是所有容器的名称前缀。可以使用配置选项 swift_store_multiple_containers_seed 来控制单/multiple 容器的使用。</p> <p>使用多个容器时，容器将通过为此配置选项设置的值命名，并将镜像 UUID 的第一个 N chars 指定为以下划线分隔的后缀（其中 N 由 swift_store_multiple_containers_seed 指定）。</p> <p>示例：如果 seed 设为 3，swift_store_container = glance，则带有 UUID fdae39a1-bac5-4238-aba4-69bcc726e848 的一个镜像将被放入容器 glance_fda。在创建容器名称时包括在 UUID 中的横线不会计算在字符长度的限制中，因此，当 N= 10 时，容器名称可以为 glance_fdae39a1-ba.</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 如果使用单个容器，此配置选项可以是 Glance Swift 帐户中有效 swift 容器名称的任何字符串 ● 如果使用多个容器，此配置选项可以是任何字符串，只要它满足 Swift 强制执行的容器命名规则。swift_store_multiple_containers_seed 的值也应该考虑。 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_multiple_containers_seed ● swift_store_multi_tenant ● swift_store_create_container_on_put

配置选项 = 默认值	类型	Description
swift_store_create_container_on_put = False	布尔值	<p>上传镜像时，创建容器（如果不存在）。</p> <p>在上传镜像时，如果对应的容器不存在，则会创建它，因为此配置选项被设置为 True。默认情况下，不会创建它。这个行为适用于单一和多个容器模式。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_endpoint = None	字符串值	<p>用于 Swift 后端存储的 URL 端点。</p> <p>提供代表在 Swift 存储中存储的 Glance 镜像的 URL 端点的字符串值。默认情况下，不会设置端点，并使用 auth 返回的存储 URL。使用 swift_store_endpoint 设置端点会覆盖存储 URL，并用于 Glance 镜像存储。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>URL 应该包括到的路径，但不包括容器。对象的位置通过将容器和对象附加到配置的 URL 来获取。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表到 Swift 容器的有效 URL 路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
swift_store_endpoint_type = publicURL	字符串值	<p>Swift 服务的端点类型。</p> <p>此字符串值表示用于获取 Swift 端点的端点类型。端点类型决定了用户被允许执行的操作，例如，读取和写入存储。只有在 swift_store_auth_version 大于 1 时才会使用此设置。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● publicURL ● adminURL ● internalURL <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_endpoint
swift_store_expire_soon_interval = 60	整数值	<p>定义在当前令牌到期前可以请求新令牌的窗口大小的时间（以秒为单位）。</p> <p>通常，Swift 存储驱动程序会在当前令牌过期时获取新令牌，以确保继续访问 Swift。但是，如果令牌实时过期，一些 Swift 事务（如上传镜像段）可能无法正常恢复。</p> <p>因此，通过在当前令牌过期前获取新令牌，请确保令牌不会在尝试事务前过期或接近到期。默认情况下，Swift 存储驱动程序请求新令牌 60 秒或少于当前令牌过期前的请求。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 零 ● 正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_key = None	字符串值	<p>用于针对 Swift 身份验证服务进行身份验证的验证密钥。</p>

配置选项 = 默认值	类型	Description
swift_store_large_object_chunk_size = 200	整数值	<p>镜像数据时段的最大大小（以 MB 为单位）。</p> <p>当镜像数据被分割为上传大于 Swift 集群实施的限制的镜像时，镜像数据被分成不同的片段，它们的大小不大于该配置选项指定的大小。如需了解更多详细信息，请参阅 swift_store_large_object_size。</p> <p>例如：如果 swift_store_large_object_size 为 5GB，而 swift_store_large_object_chunk_size 为 1GB，则大小为 6.2GB 的镜像将被分为 7 段，前 6 个段的大小为 1GB，第七个段的大小为 0.2GB。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数，它小于或等于 Swift 集群强制的大型对象限制。 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_large_object_size
swift_store_large_object_size = 5120	整数值	<p>大小阈值（以 MB 为单位）之后，Glance 将开始分段镜像数据。</p> <p>Swift 对单个上传对象的大小具有上限。默认情况下，这是 5GB。要上传大于这个限制的对象，对象被分成多个与清单文件绑定的较小的对象。如需更多详情，请参阅 https://docs.openstack.org/swift/latest/overview_large_objects.html</p> <p>此配置选项指定 Swift 驱动程序开始将镜像数据段到多个更小文件的大小阈值。目前，Swift 驱动程序只支持创建动态大对象。</p> <div data-bbox="815 1464 922 1599" style="float: left; margin-right: 10px;">  </div> <p>注意</p> <p>这应该通过考虑 Swift 集群强制的大型对象限制来设置。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数，它小于或等于 Swift 集群强制的大型对象限制。 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_large_object_chunk_size

配置选项 = 默认值	类型	Description
swift_store_multi_tenant = False	布尔值	<p>将镜像存储在租户的 Swift 帐户中。</p> <p>这将启用多租户存储模式，使 Glance 镜像存储在租户特定 Swift 帐户中。如果禁用，Glance 会将所有镜像存储在自己的帐户中。有关多租户存储的详情，请参考 https://wiki.openstack.org/wiki/GlanceSwiftTenantSpecificStorage</p> <p> 注意</p> <p>如果使用多租户 swift 存储，请确保不要使用 <i>swift_store_config_file</i> 选项设置 swift 配置文件。</p> <p>可能的值：</p> <ul style="list-style-type: none">• True• False <p>相关选项：</p> <ul style="list-style-type: none">• <code>swift_store_config_file</code>

配置选项 = 默认值	类型	Description
swift_store_multiple_containers_seed = 0	整数值	<p>指出用于存储镜像的容器数量。</p> <p>在使用单租户存储时，镜像可以存储在一个或多个容器中。当设置为 0 时，所有镜像将存储在一个容器中。当设置为 1 到 32 之间的整数值时，将使用多个容器来存储镜像。此配置选项将确定创建多少个容器。使用的容器总数等于 16^N，因此如果该配置选项设为 2，则使用 $16^2=256$ 容器来存储镜像。</p> <p>有关命名规则的详情，请参阅 swift_store_container。有关使用多个容器的详情，请参考 https://specs.openstack.org/openstack/glance-specs/specs/kilo/swift-store-multiple-containers.html</p> <p> 注意</p> <p>这仅在禁用 <code>swift_store_multi_tenant</code> 时使用。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 小于或等于 32 的非负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_container ● swift_store_multi_tenant ● swift_store_create_container_on_put

配置选项 = 默认值	类型	Description
swift_store_region = None	字符串值	<p>Glance 使用的 Swift 端点区域。</p> <p>提供代表 Swift 区域的字符串值，其中 Glance 可以连接到该区域进行镜像存储。默认情况下，没有设置区域。</p> <p>当 Glance 使用 Swift 作为存储后端存储具有多个端点的特定租户的镜像时，使用 swift_store_region 设置 Swift 区域将允许 Glance 连接到指定区域中的 Swift，而不是单一区域连接。</p> <p>可以为单租户和多租户存储配置此选项。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>使用 swift_store_region 设置区域是特定于租户的，只有在 租户在不同区域上有多个端点时才需要。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表有效 Swift 区域的字符串值。 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_retry_get_count = 0	整数值	<p>在请求失败前，将重试 Swift 下载的次数。</p> <p>提供一个整数值，代表镜像下载的次数，然后才能出错。默认值为 0（在失败的镜像下载时不重试）。当设置为正整数值时，swift_store_retry_get_count 确保在发送错误消息前下载失败时尝试多次下载。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 零 ● 正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
<p>swift_store_service_type = object-store</p>	<p>字符串值</p>	<p>要使用的 Swift 服务类型。</p> <p>提供代表在使用 Swift 后端存储期间存储镜像的服务类型的字符串值。默认服务类型设置为 object-store。</p> <div data-bbox="815 405 922 629" style="border: 1px solid #ccc; padding: 5px; width: fit-content;">  </div> <p>注意</p> <p>如果 swift_store_auth_version 设为 2，则此配置选项的值需要是 object-store。如果使用较新版本的 Keystone 或不同的身份验证方案，则可以修改此选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表 Swift 存储的有效服务类型的字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
<p>swift_store_ssl_compression = True</p>	<p>布尔值</p>	<p>HTTPS Swift 请求的 SSL 层压缩。</p> <p>提供布尔值，以确定是否压缩 SSL 层上镜像的 HTTPS Swift 请求。默认情况下启用压缩。</p> <p>当使用 Swift 作为 Glance 镜像存储的后端存储时，可以使用此选项设置 HTTPS Swift 请求的 SSL 层压缩。如果设置为 False，则禁用 HTTPS Swift 请求的 SSL 层压缩。禁用这个选项可能会提高已压缩格式的镜像的性能，如 qcow2。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
swift_store_use_trusts = True	布尔值	<p>对多租户 Swift 存储使用 trusts。</p> <p>此选项指示 Swift 存储在使用多租户存储时为每个 add/get 请求创建一个信任。使用信任允许 Swift 存储避免在上传或下载数据期间由身份验证令牌过期的问题。</p> <p>默认情况下，swift_store_use_trusts 被设置为 True（启用使用信任）。如果设置为 False，则用户令牌用于 Swift 连接，从而消除了创建信任的开销。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>只有在 swift_store_multi_tenant 被设置为 True 时，才会考虑此选项</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_multi_tenant
swift_store_user = None	字符串值	对 Swift 身份验证服务进行身份验证的用户。
swift_upload_buffer_dir = None	字符串值	<p>上传到 Swift 前用于缓冲镜像片段的目录。</p> <p>提供一个字符串值，代表 glance 节点上目录的绝对路径，在镜像片段上传到 swift 之前会短暂缓冲镜像段。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 这只有在配置选项 swift_buffer_on_upload 设置为 True 时才需要。 ● 应置备此目录，请记住 swift_store_large_object_chunk_size 以及给定 glance 节点可以同时上传的最大镜像数。 <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表绝对路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_buffer_on_upload ● swift_store_large_object_chunk_size

配置选项 = 默认值	类型	Description
vmware_api_retry_count = 10	整数值	<p>VMware API 重试次数。</p> <p>此配置选项指定连接相关问题或服务器 API 调用过载时必须重试 VMware ESX/VC 服务器 API 的次数。无法指定 <i>retry forever</i>。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> None
vmware_ca_file = None	字符串值	<p>CA 捆绑包文件的绝对路径。</p> <p>此配置选项可让操作员使用自定义 Certificate 授权机构文件来验证 ESX/vCenter 证书。</p> <p>如果设置了这个选项，则忽略 "vmware_insecure" 选项，指定的 CA 文件将用于验证 ESX/vCenter 服务器证书，并建立与服务器的安全连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何是 CA 文件的有效绝对路径的字符串 <p>相关选项：</p> <ul style="list-style-type: none"> vmware_insecure
vmware_datastores = None	多值	<p>用于存储镜像的数据存储。</p> <p>此配置选项指定镜像存储在 VMWare 存储后端中的数据存储。这个选项可多次指定，以指定多个数据存储。数据存储名称应在其数据中心路径后指定，用 ":" 分隔。可以在数据存储名称后赋予可选权重，再次通过 ":" 分隔来指定优先级。因此，所需格式将变为 <datacenter_path>:<datastore_name>:<optional_weight>。</p> <p>在添加镜像时，会选择权重最高的数据存储，除非已知镜像大小时没有足够的可用空间。如果未指定权重，则假定为 0，目录将被视为最后选择。如果多个数据存储具有相同的权重，则会选择一个带有最多可用空间的权重。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何格式的字符串：<datacenter_path>:<datastore_name>:<optional_weight> <p>相关选项：* None</p>

配置选项 = 默认值	类型	Description
vmware_insecure = False	布尔值	<p>设置 ESX/vCenter 服务器证书验证。</p> <p>此配置选项使用布尔值来确定是否验证 ESX/vCenter 服务器证书。如果此选项设为 True，则不会验证 ESX/vCenter 服务器证书。如果此选项设置为 False，则使用默认 CA 信任存储进行验证。</p> <p>如果设置了 "vmware_ca_file" 选项，则忽略这个选项。在这种情况下，将使用使用 "vmware_ca_file" 选项指定的文件来验证 ESX/vCenter 服务器证书。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • vmware_ca_file
vmware_server_host = None	主机地址值	<p>ESX/ESXi 或 vCenter Server 目标系统的地址。</p> <p>此配置选项设置 ESX/ESXi 或 vCenter Server 目标系统的地址。使用 VMware 存储后端时需要这个选项。地址可以包含 IP 地址(127.0.0.1)或 DNS 名称 (www.my-domain.com)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 有效的 IPv4 或 IPv6 地址 • 有效的 DNS 名称 <p>相关选项：</p> <ul style="list-style-type: none"> • vmware_server_username • vmware_server_password

配置选项 = 默认值	类型	Description
vmware_server_password = None	字符串值	<p>服务器密码。</p> <p>此配置选项使用密码通过 VMware ESX/ESXi 或 vCenter 服务器进行身份验证。使用 VMware 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串，与使用 "vmware_server_username" 选项指定的用户名对应的密码 <p>相关选项：</p> <ul style="list-style-type: none"> vmware_server_host vmware_server_username
vmware_server_username = None	字符串值	<p>服务器用户名。</p> <p>此配置选项使用用户名来通过 VMware ESX/ESXi 或 vCenter 服务器进行身份验证。使用 VMware 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串，这是具有适当权限的用户的用户名 <p>相关选项：</p> <ul style="list-style-type: none"> vmware_server_host vmware_server_password
vmware_store_image_dir = /openstack_glance	字符串值	<p>glance 镜像要存储在数据存储中的目录。</p> <p>此配置选项指定 glance 镜像存储在 VMware 数据存储中的目录的路径。如果没有设置这个选项，则存储 glance 镜像的默认目录为 openstack_glance。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何是到目录的有效路径的字符串 <p>相关选项：</p> <ul style="list-style-type: none"> None

配置选项 = 默认值	类型	Description
vmware_task_poll_interval = 5	整数值	<p>用于轮询在 VMware ESX/VC 服务器上调用的远程任务的间隔（以秒为单位）。</p> <p>此配置选项需要休眠时间（以秒为单位）轮询 on-going async 任务，作为 VMWare ESX/VC 服务器 API 调用的一部分。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> None

3.2.4. oslo_concurrency

下表概述了 `/etc/glance/glance-scrubber.conf` 文件中的 `[oslo_concurrency]` 组下可用的选项。

表 3.28. oslo_concurrency

配置选项 = 默认值	类型	Description
disable_process_locking = False	布尔值	启用或禁用进程间锁定。
lock_path = None	字符串值	用于锁定文件的目录。为安全起见，指定目录应只由运行需要锁定的进程的用户写入。默认为环境变量 <code>OSLO_LOCK_PATH</code> 。如果使用外部锁定，则必须设置锁定路径。

3.2.5. oslo_policy

下表概述了 `/etc/glance/glance-scrubber.conf` 文件中的 `[oslo_policy]` 组下可用的选项。

表 3.29. oslo_policy

配置选项 = 默认值	类型	描述
enforce_new_defaults = False	布尔值	这个选项控制在评估策略时是否使用旧的弃用默认值。如果为 True ，则不会评估旧的已弃用的默认值。这意味着，如果旧默认值允许任何现有令牌，但不允许使用新默认值，它将被禁止。我们鼓励启用此标志以及 enforce_scope 标志，以便您可以同时获得新默认值和 scope_type 的好处


配置选项 = 默认值	类型	描述
enforce_scope = False	布尔值	此选项控制在评估策略时是否强制执行范围。如果为 True ，则请求中使用的令牌范围将与被强制执行策略的 scope_types 进行比较。如果范围不匹配，则引发 InvalidScope 异常。如果为 False ，则会记录一条信息，告知操作员使用了不匹配的范围调用策略。
policy_default_rule = default	字符串值	默认规则。未找到请求的规则时强制执行。
policy_dirs = ['policy.d']	多值	存储策略配置文件的目录。它们可以相对于 config_dir 选项或绝对路径定义的搜索路径中的任何目录。由 policy_file 定义的文件必须存在，才能搜索这些目录。缺少或空目录将被忽略。
policy_file = policy.yaml	字符串值	将角色映射到给定服务权限的文件的相对或绝对路径。必须使用与配置文件设置相关的相对路径。
remote_content_type = application/x-www-form-urlencoded	字符串值	用于发送和接收基于 REST 策略检查的数据的内容类型
remote_ssl_ca_cert_file = None	字符串值	基于 REST 的策略检查的 ca cert 文件的绝对路径
remote_ssl_client_cert_file = None	字符串值	基于 REST 的策略检查的客户端证书的绝对路径
remote_ssl_client_key_file = None	字符串值	绝对路径客户端密钥文件 REST 基于策略检查
remote_ssl_verify_server_cert = False	布尔值	基于 REST 的策略检查的服务器身份验证

3.3. GLANCE-CACHE.CONF

本节包含 `/etc/glance/glance-cache.conf` 文件的选项。

3.3.1. DEFAULT

下表概述了 `/etc/glance/glance-cache.conf` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
<p>allow_additional_image_properties = True</p>	<p>布尔值</p>	<p>允许用户为镜像添加额外的/自定义属性。</p> <p>Glance 定义在每个镜像上显示的标准属性集合（在其架构中）。这些属性也称为基本属性。除了这些属性，Glance 允许用户向镜像添加自定义属性。它们被称为附加属性。</p> <p>默认情况下，此配置选项被设置为 True，允许用户添加额外的属性。可以通过 image_property_quota 配置选项控制可添加到镜像中的额外属性数量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • image_property_quota <p>弃用自：U ssuri</p> <p>reason: 这个选项是多余的。通过 <i>image_property_quota</i> 配置选项控制自定义镜像属性的使用。这个选项计划在 Victoria 开发周期中删除。</p>
<p>api_limit_max = 1000</p>	<p>整数值</p>	<p>请求可以返回的最大结果数。</p> <p>如 limit_param_default 的帮助文本中所述，一些请求可能会返回多个结果。要返回的结果数量取决于请求中的 limit 参数或 limit_param_default 配置选项。两个情况下的值都不能大于该配置选项定义的绝对最大值。任何大于这个值的都会修剪到此处定义的最大值。</p> <div data-bbox="815 1496 922 1688" style="display: inline-block; vertical-align: top;">  </div> <p style="margin-left: 20px;">注意</p> <p style="margin-left: 20px;">把它设置为非常大的值可能会减慢数据库查询并增加响应时间。把它设置为非常低的值可能会导致用户体验不佳。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 任何正整数 <p>相关选项：</p> <ul style="list-style-type: none"> • limit_param_default

配置选项 = 默认值	类型	Description
debug = False	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。 如果设置了 log_config_append，则忽略此选项。


配置选项 = 默认值	类型	Description
digest_algorithm = sha256	字符串值	<p>用于数字签名的摘要算法。</p> <p>提供代表用于生成数字签名的摘要算法的字符串值。默认情况下使用 sha256。</p> <p>要获得您的平台上 OpenSSL 版本支持的可用算法列表，请运行以下命令：openssl list-message-digest-algorithms。示例是 <i>sha1</i>、<i>sha256</i> 和 <i>sha512</i>。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>digest_algorithm 与 Glance 的镜像签名和验证无关。它仅用于签署通用唯一标识符(UUID)，作为证书文件和密钥文件验证的一部分。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● OpenSSL 消息摘要算法标识符 <p>关系选项：</p> <ul style="list-style-type: none"> ● None
enabled_import_methods = ['glance-direct', 'web-download', 'copy-image']	列表值	<p>启用的镜像导入方法列表</p> <div style="border-left: 2px solid black; padding-left: 10px; margin-bottom: 10px;"> <p>'glance-direct', 'copy-image' and 'web-download' are enabled by default.</p> </div> <div style="border-left: 2px solid black; padding-left: 10px;"> <p>Related options: ** [DEFAULT]/node_staging_uri</p> </div>

配置选项 = 默认值	类型	Description
enforce_secure_rbac = False	布尔值	<p>根据 OpenStack 中使用的常见用户角色定义，实施 API 访问。启用此选项规范化特定于项目的读/写操作，如创建私有镜像或更新 member 角色后共享镜像的状态。它还规范了一个可用于特定于项目的 API 操作的只读变体，如在 reader 角色后列出项目中的私有镜像。</p> <p>Operator 应该有机会了解 glance 的新镜像策略，在其部署中审核分配，并使用 keystone 中的默认角色（如 admin、member 和 reader）更新权限。</p> <p>相关选项：</p> <ul style="list-style-type: none"> • [oslo_policy]/enforce_new_defaults <p>弃用自：Wallaby</p> <p>原因：已经引进了此选项，要求操作员根据常见的 RBAC 用户角色选择强制授权，这是 Wallaby 发行版本的 EXPERIMENTAL。此行为将是未来的版本中的默认设置和 STABLE，允许删除这个选项。</p>
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。


配置选项 = 默认值	类型	Description
<p>hashing_algorithm = sha512</p>	<p>字符串值</p>	<p>用于计算 <i>os_hash_value</i> 属性的安全哈希算法。</p> <p>此选项配置 Glance "multihash"，它由两个镜像属性组成：<i>os_hash_algo</i> 和 <i>os_hash_value</i>。<i>os_hash_algo</i> 将由此配置选项的值填充，当算法应用到上传或导入的镜像数据时，<i>os_hash_value</i> 将由 <i>hexdigest</i> 填充。</p> <p>该值必须是 python <i>hashlib</i> 库识别的有效安全哈希算法名称。您可以通过检查 Glance 安装中使用的库的 <i>hashlib.algorithms_available</i> 数据成员来确定这些内容。然而，为了实现互操作性，我们建议您使用由 <i>hashlib.algorithms_guaranteed</i> 数据成员提供的安全散列名称集，因为所有平台上的 <i>hashlib</i> 库都可以保证支持这些算法。因此，在本地使用 <i>hashlib</i> 的任何镜像消费者都应能够验证镜像的 <i>os_hash_value</i>。</p> <p><i>sha512</i> 的默认值是一个性能安全哈希算法。</p> <p>如果此选项配置不正确，任何尝试存储镜像数据都将失败。因此，我们建议使用默认值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● Python <i>hashlib</i> 库识别的任何安全哈希算法名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
image_cache_dir = None	字符串值	<p>镜像缓存的基础目录。</p> <p>这是缓存镜像数据的位置，并从其中提供。所有缓存的镜像都直接存储在这个目录中。此目录还包含三个子目录，即 不完整、无效 和 队列。</p> <p>incomplete 子目录是下载镜像的暂存区域。镜像首次下载到此目录。当镜像下载成功时，它会移到基础目录。但是，如果下载失败，部分下载的镜像文件将移到 无效的 子目录中。</p> <p>queue 子目录用于排队镜像以进行下载。这主要由 cache-prefetcher 使用，它可以作为定期任务调度，如 cache-pruner 和 cache-cleaner，以在使用前缓存镜像。在收到缓存镜像的请求后，Glance 将把文件放置在 'queue 目录中，其镜像 ID 为文件名。cache-prefetcher 在运行时，轮询 队列 目录中的文件，并按照创建的顺序开始下载它们。下载成功后，从 队列 目录中删除零化文件。如果下载失败，则零化文件会保留，并将在下次运行 cache-prefetcher 时重试。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● image_cache_sqlite_db

配置选项 = 默认值	类型	Description
image_cache_driver = sqlite	字符串值	<p>用于镜像缓存管理的驱动程序。</p> <p>此配置选项提供在不同镜像缓存驱动程序之间进行选择的灵活性。image-cache 驱动程序负责提供 image-cache 的基本功能，如从缓存写入/读取镜像，跟踪缓存镜像的年龄和使用，提供缓存镜像列表、缓存获取大小、用于缓存和清理缓存等。</p> <p>驱动程序的基本功能在基本类 glance.image_cache.drivers.base.Driver 中定义。所有镜像缓存驱动程序（现有驱动程序和潜在版本）都必须实现这个接口。目前可用的驱动程序有 sqlite 和 xattr。这些驱动程序主要有所不同，它们存储缓存镜像的信息：</p> <ul style="list-style-type: none"> ● sqlite 驱动程序使用 sqlite 数据库（位于本地每个 glance 节点上）来跟踪缓存镜像的使用。 ● xattr 驱动程序使用文件的扩展属性来存储此信息。它还需要一个在访问时对文件设置 atime 的文件系统。 <p>可能的值：</p> <ul style="list-style-type: none"> ● SQLite ● xattr <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
image_cache_max_size = 10737418240	整数值	<p>缓存大小上限（以字节为单位），这会清理镜像缓存。</p> <div data-bbox="815 331 922 622" style="border: 1px solid #ccc; padding: 5px; width: fit-content;">  </div> <p>注意</p> <p>这只是 cache-pruner 操作的阈值。它不是镜像缓存永远不会增长的硬限制。实际上，根据 cache-pruner 的运行频率以及缓存填满的速度，镜像缓存不能超过此处指定的大小。因此，必须小心才能适当地调度 cache-pruner 并设定这个限制。</p> <p>Glance 在下载镜像时缓存镜像。因此，随着下载数量增加，镜像缓存的大小会随时间而增长。要将缓存大小保持不管理，建议将 cache-pruner 作为定期任务运行。当缓存修剪器启动时，它会比较镜像缓存的当前大小，并在镜像缓存超过此处指定的大小时触发清理。清理后，缓存的大小应小于或等于此处指定的大小。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何非负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
image_cache_sqlite_db = cache.db	字符串值	<p>用于镜像缓存管理的 sqlite 文件数据库的相对路径。</p> <p>这是到 sqlite 文件数据库的相对路径，用于跟踪镜像缓存的年龄和使用统计信息。该路径相对于镜像缓存基础目录，由配置选项 image_cache_dir 指定。</p> <p>这是一个轻量级数据库，只有一个表。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 到 sqlite 文件数据库的有效相对路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● image_cache_dir

配置选项 = 默认值	类型	Description
image_cache_stall_time = 86400	整数值	<p>不完整的镜像保留在缓存中的时间（以秒为单位）。</p> <p>不完整的镜像是下载正在进行的镜像。详情请查看配置选项 image_cache_dir 描述。有时，由于各种原因，下载可能会挂起，未下载的镜像保留在 不完整的 目录中。此配置选项设置在清理前镜像应保留在 不完整 目录中的时间限制。当不完整的镜像花费的时间超过此处指定的时间后，将在下次运行时通过 <code>cache-cleaner</code> 删除。</p> <p>建议将 <code>cache-cleaner</code> 作为 Glance API 节点上的定期任务运行，以保持磁盘空间中不完整的镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何非负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
image_location_quota = 10	整数值	<p>镜像上允许的最大位置数。</p> <p>任何负值都解释为无限。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● None
image_member_quota = 128	整数值	<p>每个镜像的最大镜像成员数量。</p> <p>这限制了镜像可以被共享的最大用户。任何负值都解释为无限。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
image_property_quota = 128	整数值	<p>镜像上允许的最大属性数。</p> <p>这对镜像可以具有的额外属性数量强制实施上限。任何负值都解释为无限。</p> <div style="display: flex; align-items: flex-start;">  <div> <p>注意</p> <p>如果禁用附加属性，则这不会影响。请参阅 allow_additional_image_properties。</p> </div> </div> <p>相关选项：</p> <ul style="list-style-type: none"> ● allow_additional_image_properties
image_size_cap = 1099511627776	整数值	<p>用户可以上传的最大镜像大小（以字节为单位）。</p> <p>大于此处提到的大小的镜像上传会导致镜像创建失败。此配置选项默认为 1099511627776 字节(1 TiB)。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 这个值只有在仔细考虑后才会增加，且必须设置小于或等于 8 EiB (9223372036854775808)。 ● 必须谨慎考虑后端存储容量设置这个值。把它设置为非常低的值可能会导致大量镜像失败。然后，将其设置为一个非常大的值可能会导致存储更快地消耗。因此，这必须根据创建的镜像的性质和可用的存储容量来设置。 <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何小于或等于 9223372036854775808 的正数
image_tag_quota = 128	整数值	<p>镜像上允许的最大标签数。</p> <p>任何负值都解释为无限。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● None
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。

配置选项 = 默认值	类型	Description
limit_param_default = 25	整数值	<p>为请求返回的默认结果数。</p> <p>对特定 API 请求的响应（如列出镜像）可能会返回多个项目。通过在 API 请求中指定 limit 参数，可以显式控制返回的结果数量。但是，如果没有指定 limit 参数，则此配置值将用作任何 API 请求返回的默认结果数。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 此配置选项的值可能不大于 api_limit_max 指定的值。 ● 把它设置为非常大的值可能会减慢数据库查询并增加响应时间。把它设置为非常低的值可能会导致用户体验不佳。 <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● api_limit_max
log-config-append = None	字符串值	<p>日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 log-date-format）。</p>
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	<p>在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认：<code>%(default)s</code>。如果设置了 log_config_append，则忽略此选项。</p>
log-dir = None	字符串值	<p>（可选）用于相对 log_file 路径的基础目录。如果设置了 log_config_append，则忽略此选项。</p>
log-file = None	字符串值	<p>（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 use_stderr 定义的 stderr。如果设置了 log_config_append，则忽略此选项。</p>
log_rotate_interval = 1	整数值	<p>轮转日志文件前的时间长度。除非将 log_rotation_type 设置为 "interval"，否则此选项将被忽略。</p>
log_rotate_interval_type = days	字符串值	<p>轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。</p>

配置选项 = 默认值	类型	Description
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 DEBUG 时，要附加到日志消息的其他数据。 oslo_log.formatters.ContextFormatter 使用
logging_default_format_s tring = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_for mat = %(user) s %(tenant) s %(domain) s % (user_domain) s % (project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。

配置选项 = 默认值	类型	Description
metadata_encryption_key = None	字符串值	<p>用于加密存储位置元数据的 AES 密钥。</p> <p>提供代表用于加密 Glance 存储元数据的 AES 密码的字符串值。</p> <p> 注意</p> <p>要使用的 AES 密钥必须设置为随机字符串，长度为 16、24 或 32 字节。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表有效 AES 密钥的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
node_staging_uri = file:///tmp/staging/	字符串值	<p>URL 提供临时存储临时数据的位置</p> <p>此选项仅用于 Glance 内部用途。Glance 将在镜像导入过程中保存用户上传的镜像数据到 <i>暂存</i> 端点。</p> <p>这个选项不会以任何方式更改 <i>staging</i> API 端点。</p> <p> 注意</p> <p>不建议使用与 [task]/work_dir 相同的路径</p> <p> 注意</p> <p>file:/// <absolute-directory-path > 是现在唯一支持的 api_image_import 流。</p> <p> 注意</p> <p>暂存路径必须位于所有 Glance API 节点可用的共享文件系统上。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 以 <i>file://</i> 开始的字符串，后跟绝对 FS 路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● [task]/work_dir
publish_errors = False	布尔值	启用或禁用错误事件的发布。

配置选项 = 默认值	类型	Description
pydev_worker_debug_host = None	主机地址值	<p>pydev 服务器的主机地址。</p> <p>提供代表用于调试的 pydev 服务器的主机名或 IP 的字符串值。pydev 服务器侦听此地址上的调试连接，促进 Glance 中的远程调试。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效主机名 ● 有效的 IP 地址 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
pydev_worker_debug_port = 5678	端口值	<p>pydev 服务器将侦听的端口号。</p> <p>提供将 pydev 服务器绑定到的端口号。pydev 进程接受此端口上的调试连接，并有助于在 Glance 中进行远程调试。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效端口号 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。

配置选项 = 默认值	类型	Description
<p>show_image_direct_url = False</p>	<p>布尔值</p>	<p>在返回镜像时显示直接镜像位置。</p> <p>此配置选项指示在向用户返回镜像详情时是否显示直接镜像位置。直接镜像位置是镜像数据存储在后端存储中的位置。此镜像位置显示在镜像属性 direct_url 下。</p> <p>当镜像存在多个镜像位置时，会根据配置选项 location_strategy 指示的位置策略显示最佳位置。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 显示镜像位置可能会显示 GRAVE SECURITY RISK，因为镜像位置有时可以包含凭证。因此，默认设置为 False。使用 EXTREME CAUTION 把它设置为 True，仅 IF 您知道自己正在做什么！ ● 如果运算符希望避免向用户显示任何镜像位置，则此选项和 show_multiple_locations MUST 都设为 False。 <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● show_multiple_locations ● location_strategy

配置选项 = 默认值	类型	Description
show_multiple_locations = False	布尔值	<p>显示返回镜像时的所有镜像位置。</p> <p>此配置选项指示在向用户返回镜像详情时是否显示所有镜像位置。当镜像存在多个镜像位置时，根据配置 opt location_strategy 指示的位置策略进行排序。镜像位置显示在镜像属性 位置 下。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 显示镜像位置可能会显示 GRAVE SECURITY RISK，因为镜像位置有时可以包含凭证。因此，默认设置为 False。使用 EXTREME CAUTION 把它设置为 True，仅 IF 您知道自己正在做什么！ ● 如需更多信息，请参阅 https://wiki.openstack.org/wiki/OSSN/OSSN-0065。 ● 如果运算符希望避免向用户显示任何镜像位置，则此选项和 show_image_direct_url MUST 都被设置为 False。 <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● show_image_direct_url ● location_strategy <p>弃用自：Newton</p> <p>*reason:*Use of option（自 Newton 起已弃用）是一个安全风险，当我们找出满足目前需要它的用例后，将删除它。之前宣布可使用策略以更高的粒度实现相同的功能不正确。您无法在当前时间通过策略配置临时解决此选项，但这是我们认为该修复将采取的方向。请留意 Glance 发行注记，以保持在这个问题的最新进度。</p>
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。

配置选项 = 默认值	类型	Description
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 <code>log_config_append</code> ，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 <code>log_config_append</code> ，则忽略此选项。
user_storage_quota = 0	字符串值	<p>每个租户的最大镜像存储量。</p> <p>这会强制对所有存储中的所有租户镜像使用的累积存储进行上限。这是每个租户的限制。</p> <p>此配置选项的默认单位是字节。但是，可以使用区分大小写的字面 B、KB、MB、GB 和 TB 指定存储单元，分别代表 Bytes、KiloBytes、MegaBytes、GigaBytes 和 TeraBytes。请注意，值和单元之间不应有任何空格。值 0 表示没有配额强制。负值无效，并导致错误。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 一个字符串，它是一个非负整数的有效串联，代表存储值，以及代表上面提到的存储单元的可选字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> None
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 <code>log_file</code> 选项并使用 Linux 平台时，才有意义。如果设置了 <code>log_config_append</code> ，则忽略此选项。

配置选项 = 默认值	类型	Description
worker_self_reference_url = None	字符串值	<p>此 worker 的 URL。</p> <p>如果设置了此设置，其他 glance 工作程序将了解如何直接联系此 worker。对于镜像导入，单个 worker 分阶段镜像和其他 worker 需要能够将导入请求代理到正确的请求。</p> <p>如果未设置，这被视为 public_endpoint，这通常会设置为所有 worker 上的相同值，从而有效地禁用代理行为。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 可以从其他 worker 访问此 worker 的 URL <p>相关选项：</p> <ul style="list-style-type: none"> ● public_endpoint

3.3.2. glance_store

下表概述了 `/etc/glance/glance-cache.conf` 文件中的 `[glance_store]` 组下可用的选项。

表 3.30. glance_store

配置选项 = 默认值	类型	Description
cinder_api_insecure = False	布尔值	<p>允许对 cinder 执行不安全的 SSL 请求。</p> <p>如果此选项被设置为 True，则使用 cinder_ca_certificates_file 选项指定的 CA 证书文件验证 HTTPS 端点连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_ca_certificates_file

配置选项 = 默认值	类型	Description
cinder_ca_certificates_file = None	字符串值	<p>用于 Cinder 客户端请求的 CA 证书文件的位置。</p> <p>指定的 CA 证书文件（如果设置）用于通过 HTTPS 端点验证 Cinder 连接。如果端点是 HTTP，则忽略这个值。cinder_api_insecure 必须设置为 True 以启用验证。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● ca 证书文件的路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_api_insecure
cinder_catalog_info = volumev3::publicURL	字符串值	<p>在服务目录中查找 cinder 时匹配的信息。</p> <p>如果没有设置 cinder_endpoint_template，并且任何 cinder_store_auth_address，cinder_store_user_name，cinder_store_project_name，cinder_store_password 没有设置，cinder 存储将使用此信息从当前上下文中的服务目录中查找 Cinder 端点。如果设置，将考虑 cinder_os_region_name 来获取适当的端点。</p> <p>服务目录可以通过 openstack catalog list 命令列出。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 应指定以下形式的字符串：< service_type>:<service_name>:<interface> At least service_type 和 interface。 service_name 可以省略。 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_os_region_name ● cinder_endpoint_template ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password

配置选项 = 默认值	类型	Description
cinder_endpoint_template = None	字符串值	<p>使用 cinder 端点的模板覆盖服务目录查找。</p> <p>设置此选项时，这个值用于生成 cinder 端点，而不是从服务目录查找。如果指定了 cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, 和 cinder_store_password, 则忽略这个值。</p> <p>如果设置了此配置选项，则忽略 cinder_catalog_info。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● cinder 端点的 URL 模板字符串，其中 %%(tenant) 替换为当前的租户（项目）名称。例如： http://cinder.openstack.example.org/v2/%%(tenant)s <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name ● cinder_store_password ● cinder_catalog_info
cinder_enforce_multipath = False	布尔值	<p>如果设置为 True，则当 multipathd 未运行时，镜像传输的卷附加将中止。否则，它将回退到单一路径。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True 或 False <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_use_multipath
cinder_http_retries = 3	整数值	<p>失败 http 调用的 cinderclient 重试次数。</p> <p>当调用被任何错误失败时，cinderclient 会在睡眠几秒钟后重试调用。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
cinder_mount_point_base = /var/lib/glance/mnt	字符串值	<p>NFS 卷挂载在 glance 节点上的目录。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表挂载点的绝对路径的字符串。
cinder_os_region_name = None	字符串值	<p>从服务目录查找 cinder 服务的区域名称。</p> <p>这仅在使用 cinder_catalog_info 来确定端点时使用。如果设置，则此节点的 cinder 端点的查找将过滤到指定的区域。当目录中列出了多个区域时，它很有用。如果没有设置，则会从每个区域查找端点。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 是有效区域名称的字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_catalog_info
cinder_state_transition_timeout = 300	整数值	<p>等待 cinder 卷转换完成的时间（以秒为单位）。</p> <p>当创建、删除或附加到 glance 节点以读/写卷数据时，卷的状态将改变。例如，新创建的卷状态会在 创建 过程完成后变为 available。这将指定等待状态更改的最长时间。如果在等待时发生超时，或者状态更改为非预期的值（如 错误），则镜像创建会失败。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
cinder_store_auth_addresses = None	字符串值	<p>cinder 身份验证服务正在侦听的地址。</p> <p>当指定所有 cinder_store_auth_address, cinder_store_user_name, cinder_store_project_name, 和 cinder_store_password 选项时, 指定的值始终用于身份验证。这有助于通过将镜像服务存储在特定于镜像服务的项目/租户中, 从用户隐藏镜像卷。它还允许用户在 glance 的 ACL 控制下与其他项目共享镜像卷。</p> <p>如果没有设置任何一个选项, 则 cinder 端点会从服务目录中查找, 并使用当前上下文的用户和项目。</p> <p>可能的值 :</p> <ul style="list-style-type: none"> ● 有效的身份验证服务地址, 例如 : http://openstack.example.org/identity/v2.0 <p>相关选项 :</p> <ul style="list-style-type: none"> ● cinder_store_user_name ● cinder_store_password ● cinder_store_project_name
cinder_store_password = None	字符串值	<p>用于针对 Cinder 进行身份验证的用户密码。</p> <p>这必须与以下所有相关选项一起使用。如果未指定其中任何一个, 则使用当前上下文的用户。</p> <p>可能的值 :</p> <ul style="list-style-type: none"> ● 由 cinder_store_user_name 指定的用户的有效密码 <p>相关选项 :</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_project_name

配置选项 = 默认值	类型	Description
cinder_store_project_name = None	字符串值	<p>镜像卷的项目名称存储在 cinder 中。</p> <p>如果没有设置此配置选项，则使用当前上下文中的项目。</p> <p>这必须与以下所有相关选项一起使用。如果未指定其中任何一个，则使用当前上下文的项目。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效项目名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_user_name ● cinder_store_password
cinder_store_user_name = None	字符串值	<p>要针对 cinder 进行身份验证的用户名。</p> <p>这必须与以下所有相关选项一起使用。如果未指定其中任何一个，则使用当前上下文的用户。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的用户名 <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_store_auth_address ● cinder_store_password ● cinder_store_project_name
cinder_use_multipath = False	布尔值	<p>在部署中支持或不支持识别 multipath 的标记。</p> <p>如果不支持多路径，则将其设置为 False。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True 或 False <p>相关选项：</p> <ul style="list-style-type: none"> ● cinder_enforce_multipath

配置选项 = 默认值	类型	Description
<p>cinder_volume_type = None</p>	<p>字符串值</p>	<p>在 cinder 中用于创建卷的卷类型。</p> <p>有些 Cinder 后端可以有几个卷类型来优化存储使用。添加此选项可让操作器在 cinder 中选择特定卷类型，该类型可以针对镜像进行了优化。</p> <p>如果没有设置，则 cinder 配置中指定的默认卷类型将用于创建卷。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 来自 cinder 的有效卷类型 <p>相关选项：</p> <ul style="list-style-type: none"> ● None <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>您不能使用与 NFS 后端关联的加密 volume_type。当 glance_store 试图写入或访问存储在该卷中的镜像数据时，存储在 NFS 后端上的加密卷将引发异常。请参考 Cinder 管理员确定适当的 volume_type。</p> </div> </div>

配置选项 = 默认值	类型	Description
default_store = file	字符串值	<p>用于存储镜像的默认方案。</p> <p>提供代表用于存储镜像的默认方案的字符串值。如果没有设置，Glance 使用 file 作为默认方案来使用文件存储镜像。</p> <div data-bbox="815 439 922 600" style="float: left; margin-right: 10px;"> </div> <p>注意</p> <p>为这个配置选项提供的值必须是使用 stores 配置选项注册的存储的有效方案。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● file ● filesystem ● http ● https ● swift ● swift+http ● swift+https ● swift+config ● rbd ● cinder ● vsphere ● s3 <p>相关选项：</p> <ul style="list-style-type: none"> ● 存储 <p>弃用自：Rocky</p> <p>reason: 这个选项根据新配置选项 "default_backend" 弃用，它的行为与 "default_store" 配置选项类似。</p> <p>计划在 U 开发周期中移除此选项。</p>

配置选项 = 默认值	类型	Description
default_swift_reference = ref1	字符串值	<p>引用默认 Swift 帐户/备份存储参数。</p> <p>提供代表使用 swift 帐户/反馈存储所需默认参数集合的引用的字符串值。此配置选项的默认引用值为 <i>ref1</i>。此配置选项解引用参数，并有助于每次添加新镜像时 Swift 存储后端中的镜像存储。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
filesystem_store_chunk_size = 65536	整数值	<p>块大小，以字节为单位。</p> <p>读取或写入镜像文件时使用的块大小。增加这个值可以提高吞吐量，但在处理大量请求时，它也可能略微增加内存用量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
<p>filesystem_store_datadir = /var/lib/glance/images</p>	<p>字符串值</p>	<p>文件系统后端存储写入镜像的目录。</p> <p>启动时，Glance 会创建该目录（如果不存在），并验证在其下运行 glance-api 的用户的写入访问权限。如果写入访问不可用，则引发 BadStoreConfiguration 异常，并且文件系统存储可能无法添加新镜像。</p> <div data-bbox="815 510 922 824" style="border: 1px solid gray; padding: 5px; width: fit-content;">  </div> <p>注意</p> <p>此目录仅在将文件系统存储用作存储后端时使用。filesystem_store_datadir 或 filesystem_store_datadirs 选项必须在 glance-api.conf 中指定。如果指定了这两个选项，则会引发 BadStoreConfiguration，并且文件系统存储可能无法添加新镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 到目录的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● filesystem_store_datadirs ● filesystem_store_file_perm

配置选项 = 默认值	类型	Description
filesystem_store_datadirs = None	多值	<p>文件系统后端存储写入镜像的目录及其优先级列表。</p> <p>文件系统存储可以配置为将镜像存储在多个目录中，而不是使用由 filesystem_store_datadir 配置选项指定的单个目录。使用多个目录时，可以为每个目录赋予一个可选优先级，以指定应使用它们的首选项顺序。priority 是一个整数，它将与目录路径连接，冒号越高，代表优先级更高。当两个目录具有相同的优先级时，会使用具有最多可用空间的目录。如果没有指定优先级，则默认为零。</p> <p>有关使用多个存储目录配置文件系统存储的更多信息，请参阅 https://docs.openstack.org/glance/latest/configuration/configuring.html</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 40px; height: 40px; border: 1px solid black; margin-right: 10px;">  </div> <div> <p>注意</p> <p>此目录仅在将文件系统存储用作存储后端时使用。filesystem_store_datadir 或 filesystem_store_datadirs 选项必须在 glance-api.conf 中指定。如果指定了这两个选项，则会引发 BadStoreConfiguration，并且文件系统存储可能无法添加新镜像。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 以下格式的字符串列表： <ul style="list-style-type: none"> ○ <a valid directory path>:<optional integer priority> <p>相关选项：</p> <ul style="list-style-type: none"> ● filesystem_store_datadir ● filesystem_store_file_perm

配置选项 = 默认值	类型	Description
filesystem_store_file_permission = 0	整数值	<p>镜像文件的文件访问权限。</p> <p>为镜像数据设置预期的文件访问权限。这提供了一种方式，可以让其他服务（如 Nova）直接使用文件系统存储中的镜像。运行要被授予访问权限的服务的用户，可以成为拥有创建文件的组的成员。为这个配置选项分配一个小于或等于 0 的值表示没有对默认权限进行任何更改。这个值将被解码为八进制数。</p> <p>如需更多信息，请参阅文档 https://docs.openstack.org/glance/latest/configuration/configuring.html</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的文件访问权限 ● 零 ● 任何负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
filesystem_store_metadata_file = None	字符串值	<p>文件系统存储元数据文件。</p> <p>包含与文件系统存储关联的任何位置返回元数据的文件路径。设置此选项后，它仅用于之后创建的新镜像 - 以前现有镜像不受影响。</p> <p>该文件必须包含有效的 JSON 对象。该对象应包含键 id 和 mountpoint。两个键的值都应该是一个字符串。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 存储元数据文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
filesystem_thin_provisioning = False	布尔值	<p>在这个后端中启用或不精简置备。</p> <p>这个配置选项启用了在文件系统中没有真正写入 null 字节序列的功能，文件系统会自动解释为 null 字节，且不会实际消耗您的存储。启用此功能也会加快镜像上传速度，并节省后端中的空间，因为不会通过网络发送 null 字节序列。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • None
http_proxy_information = {}	dict 值	<p>用于连接到远程服务器的 http/https 代理信息。</p> <p>这个配置选项指定应该用来连接到远程服务器的 http/https 代理信息。代理信息应该是方案和代理的键值对，如 http:10.0.0.1:3128。您还可以通过使用逗号分隔键值对（如 http:10.0.0.1:3128, https:10.0.0.1:1080）来指定多个方案的代理。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 如前文所述，以逗号分隔的 scheme:proxy 对列表 <p>相关选项：</p> <ul style="list-style-type: none"> • None
https_ca_certificates_file = None	字符串值	<p>CA 捆绑包文件的路径。</p> <p>此配置选项可让操作员使用自定义证书颁发机构文件来验证远程服务器证书。如果设置了这个选项，则忽略 https_insecure 选项，指定的 CA 文件将用于验证服务器证书并建立到服务器的安全连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • CA 文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> • https_insecure

配置选项 = 默认值	类型	Description
https_insecure = True	布尔值	<p>设置远程服务器证书的验证。</p> <p>此配置选项使用布尔值来确定是否验证远程服务器证书。如果设置为 True，则不会验证远程服务器证书。如果选项设为 False，则使用默认 CA 信任存储进行验证。</p> <p>如果设置了 https_ca_certificates_file，则忽略这个选项。然后，使用 https_ca_certificates_file 选项指定的文件验证远程服务器证书。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • https_ca_certificates_file
rados_connect_timeout = 0	整数值	<p>连接到 Ceph 集群的超时值。</p> <p>此配置选项在连接到 Ceph 集群时所用的超时值（以秒为单位）。它会设置在关闭连接前等待 glance-api 的时间。这可防止在连接到 RBD 时出现 glance-api 挂起。如果此选项的值设为小于或等于 0，则不设置超时，并且使用默认的 librados 值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 任何整数值 <p>相关选项：</p> <ul style="list-style-type: none"> • None

配置选项 = 默认值	类型	Description
`rbd_store_ceph_conf = `	字符串值	<p>Ceph 配置文件路径。</p> <p>此配置选项指定要使用的 Ceph 配置文件的路径。如果此选项的值未由用户设置或设置为空字符串，librados 将按顺序搜索默认的 Ceph 配置文件位置来读取标准 ceph.conf 文件。详情请参阅 Ceph 文档。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>如果使用 Cephx 身份验证，此文件应该在 client.<USER> 部分中包含对正确密钥环的引用</p> </div> </div> <p>注意 2：如果您将此选项留空（默认），使用的实际 Ceph 配置文件可能会根据使用 librados 的版本而有所不同。如果您知道哪个配置文件生效非常重要，您可以在此处使用这个选项指定该文件。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 配置文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● rbd_store_user
rbd_store_chunk_size = 8	整数值	<p>大小（以 MB 为单位）将 RADOS 镜像块到其中。</p> <p>提供代表 Glance 镜像的大小（以 MB 为单位）的整数值。默认块大小为 8MB。为获得最佳性能，该值应该是 2 的指数。</p> <p>当 Ceph 的 RBD 对象存储系统用作存储 Glance 镜像的存储后端时，镜像将使用此选项集中到设置的大小的对象中。然后，这些块的对象存储在用于 Glance 的分布式块数据存储中。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
rbd_store_pool = images	字符串值	<p>存储镜像的 RADOS 池。</p> <p>当 RBD 用作存储 Glance 镜像的存储后端时，镜像通过将对象（镜像区块）的逻辑分组存储到 池中。每个池使用它可以包含的放置组数量来定义。使用的默认池是 <i>images</i>。</p> <p>有关 RBD 存储后端的更多信息，请参阅： http://ceph.com/planet/how-data-is-stored-in-ceph-cluster/</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效池名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
rbd_store_user = None	字符串值	<p>RADOS 用户以以下身份进行身份验证：</p> <p>此配置选项在 RADOS 用户进行身份验证，以如下方式进行身份验证：这仅在启用 RADOS 身份验证时且仅在用户使用 Cephx 身份验证时才需要。如果此选项的值未由用户设置或设置为 None，则会选择一个默认值，该值将基于 <code>rbd_store_ceph_conf</code> 中的 <code>client.</code> 部分。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的 RADOS 用户 <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>rbd_store_ceph_conf</code>
rbd_thin_provisioning = False	布尔值	<p>在这个后端中启用或不精简置备。</p> <p>此配置选项允许在 RBD 后端上实际写入 null 字节序列的功能，Ceph 将自动显示的漏洞被 Ceph 解释为 null 字节，且不会实际消耗您的存储。启用此功能也会加快镜像上传速度，并节省后端中的空间，因为不会通过网络发送 null 字节序列。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
rootwrap_config = /etc/glance/rootwrap.conf	字符串值	<p>用于以 root 身份运行命令的 rootwrap 配置文件的路径。</p> <p>cinder 存储需要 root 特权才能运行镜像卷（用于连接 iSCSI/FC 卷并读取/写入卷数据等）。配置文件应允许 cinder 存储和 os-brick 库所需的命令。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● rootwrap 配置文件的路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
s3_store_access_key = None	字符串值	<p>S3 查询令牌访问密钥。</p> <p>此配置选项使用访问密钥来与 Amazon S3 或 S3 兼容存储服务器进行身份验证。使用 S3 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何字符串值，这是具有适当特权的用户的访问密钥 <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_host ● s3_store_secret_key
s3_store_bucket = None	字符串值	<p>用于存储 Glance 数据的 S3 存储桶。</p> <p>此配置选项指定 glance 镜像存储在 S3 中的位置。如果 s3_store_create_bucket_on_put 设为 true，它将自动创建，即使存储桶不存在。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_create_bucket_on_put ● s3_store_bucket_url_format

配置选项 = 默认值	类型	Description
s3_store_bucket_url_format = auto	字符串值	<p>用于确定对象的 S3 调用格式。</p> <p>此配置选项使用访问模型，用于在 S3 存储桶中指定对象地址。</p> <p>注意：在 路径- 样式中，对象的端点类似于 https://s3.amazonaws.com/bucket/example.img。在 虚拟- 样式中，对象的端点类似于 https://bucket.s3.amazonaws.com/example.img。如果您没有在存储桶名称中遵循 DNS 命名惯例，您可以使用路径风格获取对象，但不能以虚拟样式获取对象。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● auto、虚拟 或 path的任何字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_bucket
s3_store_create_bucket_on_put = False	布尔值	<p>确定 S3 是否应该创建新存储桶。</p> <p>此配置选项使用布尔值来指示 Glance 是否应该向 S3 创建新存储桶（如果不存在）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何布尔值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
s3_store_host = None	字符串值	<p>S3 服务器侦听的主机。</p> <p>此配置选项设置 S3 或 S3 兼容存储服务器的主机。使用 S3 存储后端时需要这个选项。主机可以包含 DNS 名称（如 s3.amazonaws.com、my-object-storage.com）或 IP 地址(127.0.0.1)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 有效的 DNS 名称 ● 有效的 IPv4 地址 <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_access_key ● s3_store_secret_key
s3_store_large_object_chunk_size = 10	整数值	<p>在上传部分时，S3 应该使用什么多部分上传部分大小（以 MB 为单位）。</p> <p>这个配置选项使用镜像分割大小（以 MB 为单位）进行多部分上传。</p> <p>注意：您只能分成 10,000 个镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数值（必须大于或等于 5M） <p>相关选项：</p> <ul style="list-style-type: none"> ● s3_store_large_object_size ● s3_store_thread_pools

配置选项 = 默认值	类型	Description
s3_store_large_object_size = 100	整数值	<p>S3 启动镜像文件的大小应该以 MB 为单位，并在 S3 中进行多部分上传。</p> <p>此配置选项采用以 MB 为单位的阈值，以确定是否将镜像上传到 S3，还是将其分割（多部分上传）。</p> <p>注意：您只能分成 10,000 个镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_large_object_chunk_size s3_store_thread_pools
s3_store_secret_key = None	字符串值	<p>S3 查询令牌 secret 密钥。</p> <p>此配置选项使用 secret 密钥来与 Amazon S3 或 S3 兼容存储服务器进行身份验证。使用 S3 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串值，它是与使用 s3_store_host 选项指定的访问密钥对应的 secret 键 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_host s3_store_access_key
s3_store_thread_pools = 10	整数值	<p>在 S3 中执行多部分上传的线程池数量。</p> <p>此配置选项在执行多部分上传时取线程池的数量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> s3_store_large_object_size s3_store_large_object_chunk_size

配置选项 = 默认值	类型	Description
stores = ['file', 'http']	列表值	<p>启用 Glance 存储的列表。</p> <p>注册存储后端，以逗号分隔列表的形式用于存储磁盘镜像。为使用 Glance 存储磁盘镜像而启用的默认存储是 file 和 http。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个以逗号分隔的列表，其中包括： <ul style="list-style-type: none"> ○ file ○ http ○ swift ○ rbd ○ cinder ○ vmware ○ s3 <p>相关选项：</p> <ul style="list-style-type: none"> ● default_store <p>弃用自： Rocky</p> <p>reason: 这个选项根据新配置选项 "enabled_backends" 弃用，这有助于配置不同方案的多个后端存储。</p> <p>计划在 U 开发周期中移除此选项。</p>

配置选项 = 默认值	类型	Description
swift_buffer_on_upload = False	布尔值	<p>上传到 Swift 前缓冲区镜像段。</p> <p>提供布尔值，以指示 Glance 是否应该在上传到 swift 时将镜像数据缓冲到磁盘。这可让 Glance 恢复出错时上传。</p> <p>注意：当启用这个选项时，应该非常小心，因为这会增加 API 节点上的磁盘用量。请注意，根据文件系统的配置方式，用于缓冲的磁盘空间可能会减少 glance 镜像缓存可用的实际磁盘空间。磁盘利用率将根据以下等情况设置：</p> <p>(swift_store_large_object_chunk_size * workers * 1000)</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • swift_upload_buffer_dir
swift_store_admin_tenants = []	列表值	<p>将被授予 admin 访问权限的租户列表。</p> <p>这是租户列表，将在多租户模式下授予 Glance 创建的所有 Swift 容器的读取/写入访问权限。默认值为空列表。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 以逗号分隔的字符串列表，代表 Keystone 项目/租户的 UUID <p>相关选项：</p> <ul style="list-style-type: none"> • None
swift_store_auth_address = None	字符串值	Swift 身份验证服务侦听的端口。

配置选项 = 默认值	类型	Description
swift_store_auth_insecure = False	布尔值	<p>设置服务器证书验证。</p> <p>此布尔值决定了是否验证服务器证书。如果此选项设置为 True，则 swiftclient 在验证时不会检查有效的 SSL 证书。如果选项设为 False，则使用默认 CA 信任存储进行验证。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • swift_store_cacert
swift_store_auth_version = 2	字符串值	<p>要使用的身份验证服务的版本。对于 keystone 有效版本是 2 和 3，对于 swauth 和 rackspace 为 1（已弃用）。</p>
swift_store_cacert = None	字符串值	<p>CA 捆绑包文件的路径。</p> <p>此配置选项可让操作员指定自定义证书颁发机构文件的路径，以便在连接到 Swift 时进行 SSL 验证。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • CA 文件的有效路径 <p>相关选项：</p> <ul style="list-style-type: none"> • swift_store_auth_insecure

配置选项 = 默认值	类型	Description
swift_store_config_file = None	字符串值	<p>包含 swift 帐户配置的文件的路径。</p> <p>包括一个字符串值，代表配置文件的路径，该文件对配置的每个 Swift 帐户/反馈存储的引用。默认情况下，未指定文件路径，并且禁用自定义 Swift 引用。强烈建议配置这个选项，同时将 Swift 存储后端用于镜像存储，因为它避免存储数据库中的凭证。</p> <p> 注意</p> <p>如果将 swift_store_multi_tenant 设为 True，请不要配置此选项。</p> <p>可能的值：</p> <ul style="list-style-type: none">● 代表 glance-api 节点上的绝对路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none">● swift_store_multi_tenant

配置选项 = 默认值	类型	Description
<p>swift_store_container = glance</p>	<p>字符串值</p>	<p>用于存储多个容器的 images/name 前缀的单一容器名称</p> <p>当单个容器用于存储镜像时，此配置选项将指示 Glance 帐户内用于存储所有镜像的容器。当使用多个容器来存储镜像时，这将是所有容器的名称前缀。可以使用配置选项 swift_store_multiple_containers_seed 来控制单/multiple 容器的使用。</p> <p>使用多个容器时，容器将通过为此配置选项设置的值命名，并将镜像 UUID 的第一个 N chars 指定为以下划线分隔的后缀（其中 N 由 swift_store_multiple_containers_seed 指定）。</p> <p>示例：如果 seed 设为 3，swift_store_container = glance，则带有 UUID fdae39a1-bac5-4238-aba4-69bcc726e848 的一个镜像将被放入容器 glance_fda。在创建容器名称时包括在 UUID 中的横线不会计算在字符长度的限制中，因此，当 N= 10 时，容器名称可以为 glance_fdae39a1-ba..</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 如果使用单个容器，此配置选项可以是 Glance Swift 帐户中有效 swift 容器名称的任何字符串 ● 如果使用多个容器，此配置选项可以是任何字符串，只要它满足 Swift 强制执行的容器命名规则。swift_store_multiple_containers_seed 的值也应该考虑。 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_multiple_containers_seed ● swift_store_multi_tenant ● swift_store_create_container_on_put

配置选项 = 默认值	类型	Description
swift_store_create_container_on_put = False	布尔值	<p>上传镜像时，创建容器（如果不存在）。</p> <p>在上传镜像时，如果对应的容器不存在，则会创建它，因为此配置选项被设置为 True。默认情况下，不会创建它。这个行为适用于单一和多个容器模式。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • None
swift_store_endpoint = None	字符串值	<p>用于 Swift 后端存储的 URL 端点。</p> <p>提供代表在 Swift 存储中存储的 Glance 镜像的 URL 端点的字符串值。默认情况下，不会设置端点，并使用 auth 返回的存储 URL。使用 swift_store_endpoint 设置端点会覆盖存储 URL，并用于 Glance 镜像存储。</p> <p> 注意</p> <p>URL 应该包括到的路径，但不包括容器。对象的位置通过将容器和对象附加到配置的 URL 来获取。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 代表到 Swift 容器的有效 URL 路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> • None

配置选项 = 默认值	类型	Description
swift_store_endpoint_type = publicURL	字符串值	<p>Swift 服务的端点类型。</p> <p>此字符串值表示用于获取 Swift 端点的端点类型。端点类型决定了用户被允许执行的操作，例如，读取和写入存储。只有在 swift_store_auth_version 大于 1 时才会使用此设置。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● publicURL ● adminURL ● internalURL <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_endpoint
swift_store_expire_soon_interval = 60	整数值	<p>定义在当前令牌到期前可以请求新令牌的窗口大小的时间（以秒为单位）。</p> <p>通常，Swift 存储驱动程序会在当前令牌过期时获取新令牌，以确保继续访问 Swift。但是，如果令牌实时过期，一些 Swift 事务（如上传镜像段）可能无法正常恢复。</p> <p>因此，通过在当前令牌过期前获取新令牌，请确保令牌不会在尝试事务前过期或接近到期。默认情况下，Swift 存储驱动程序请求新令牌 60 秒或少于当前令牌过期前的请求。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 零 ● 正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_key = None	字符串值	<p>用于针对 Swift 身份验证服务进行身份验证的验证密钥。</p>

配置选项 = 默认值	类型	Description
swift_store_large_object_chunk_size = 200	整数值	<p>镜像数据时段的最大大小（以 MB 为单位）。</p> <p>当镜像数据被分割为上传大于 Swift 集群实施的限制的镜像时，镜像数据被分成不同的片段，它们的大小不大于该配置选项指定的大小。如需了解更多详细信息，请参阅 swift_store_large_object_size。</p> <p>例如：如果 swift_store_large_object_size 为 5GB，而 swift_store_large_object_chunk_size 为 1GB，则大小为 6.2GB 的镜像将被分为 7 段，前 6 个段的大小为 1GB，第七个段的大小为 0.2GB。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数，它小于或等于 Swift 集群强制的大型对象限制。 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_large_object_size
swift_store_large_object_size = 5120	整数值	<p>大小阈值（以 MB 为单位）之后，Glance 将开始分段镜像数据。</p> <p>Swift 对单个上传对象的大小具有上限。默认情况下，这是 5GB。要上传大于这个限制的对象，对象被分成多个与清单文件绑定的较小的对象。如需更多详情，请参阅 https://docs.openstack.org/swift/latest/overview_large_objects.html</p> <p>此配置选项指定 Swift 驱动程序开始将镜像数据段到多个更小文件的大小阈值。目前，Swift 驱动程序只支持创建动态大对象。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>这应该通过考虑 Swift 集群强制的大型对象限制来设置。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个正整数，它小于或等于 Swift 集群强制的大型对象限制。 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_large_object_chunk_size

配置选项 = 默认值	类型	Description
swift_store_multi_tenant = False	布尔值	<p>将镜像存储在租户的 Swift 帐户中。</p> <p>这将启用多租户存储模式，使 Glance 镜像存储在租户特定 Swift 帐户中。如果禁用，Glance 会将所有镜像存储在自己的帐户中。有关多租户存储的详情，请参考 https://wiki.openstack.org/wiki/GlanceSwiftTenantSpecificStorage</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>如果使用多租户 swift 存储，请确保不要使用 <code>swift_store_config_file</code> 选项设置 swift 配置文件。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>swift_store_config_file</code>

配置选项 = 默认值	类型	Description
swift_store_multiple_containers_seed = 0	整数值	<p>指出用于存储镜像的容器数量。</p> <p>在使用单租户存储时，镜像可以存储在一个或多个容器中。当设置为 0 时，所有镜像将存储在一个容器中。当设置为 1 到 32 之间的整数值时，将使用多个容器来存储镜像。此配置选项将确定创建多少个容器。使用的容器总数等于 16^N，因此如果该配置选项设为 2，则使用 $16^2=256$ 容器来存储镜像。</p> <p>有关命名规则的详情，请参阅 swift_store_container。有关使用多个容器的详情，请参考 https://specs.openstack.org/openstack/glance-specs/specs/kilo/swift-store-multiple-containers.html</p> <p> 注意</p> <p>这仅在禁用 <code>swift_store_multi_tenant</code> 时使用。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 小于或等于 32 的非负整数 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_container ● swift_store_multi_tenant ● swift_store_create_container_on_put

配置选项 = 默认值	类型	Description
swift_store_region = None	字符串值	<p>Glance 使用的 Swift 端点区域。</p> <p>提供代表 Swift 区域的字符串值，其中 Glance 可以连接到该区域进行镜像存储。默认情况下，没有设置区域。</p> <p>当 Glance 使用 Swift 作为存储后端存储具有多个端点的特定租户的镜像时，使用 swift_store_region 设置 Swift 区域将允许 Glance 连接到指定区域中的 Swift，而不是单一区域连接。</p> <p>可以为单租户和多租户存储配置此选项。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>使用 swift_store_region 设置区域是特定于租户的，只有在 租户在不同区域上有多个端点时才需要。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表有效 Swift 区域的字符串值。 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_retry_get_count = 0	整数值	<p>在请求失败前，将重试 Swift 下载的次数。</p> <p>提供一个整数值，代表镜像下载的次数，然后才能出错。默认值为 0（在失败的镜像下载时不重试）。当设置为正整数值时，swift_store_retry_get_count 确保在发送错误消息前下载失败时尝试多次下载。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 零 ● 正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
swift_store_service_type = object-store	字符串值	<p>要使用的 Swift 服务类型。</p> <p>提供代表在使用 Swift 后端存储期间存储镜像的服务类型的字符串值。默认服务类型设置为 object-store。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>如果 swift_store_auth_version 设为 2，则此配置选项的值需要是 object-store。如果使用较新版本的 Keystone 或不同的身份验证方案，则可以修改此选项。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表 Swift 存储的有效服务类型的字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> ● None
swift_store_ssl_compression = True	布尔值	<p>HTTPS Swift 请求的 SSL 层压缩。</p> <p>提供布尔值，以确定是否压缩 SSL 层上镜像的 HTTPS Swift 请求。默认情况下启用压缩。</p> <p>当使用 Swift 作为 Glance 镜像存储的后端存储时，可以使用此选项设置 HTTPS Swift 请求的 SSL 层压缩。如果设置为 False，则禁用 HTTPS Swift 请求的 SSL 层压缩。禁用这个选项可能会提高已压缩格式的镜像的性能，如 qcow2。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● None

配置选项 = 默认值	类型	Description
swift_store_use_trusts = True	布尔值	<p>对多租户 Swift 存储使用 trusts。</p> <p>此选项指示 Swift 存储在使用多租户存储时为每个 add/get 请求创建一个信任。使用信任允许 Swift 存储避免在上传或下载数据期间由身份验证令牌过期的问题。</p> <p>默认情况下，swift_store_use_trusts 被设置为 True（启用使用信任）。如果设置为 False，则用户令牌用于 Swift 连接，从而消除了创建信任的开销。</p> <div style="display: flex; align-items: flex-start;">  <div> <p>注意</p> <p>只有在 swift_store_multi_tenant 被设置为 True 时，才会考虑此选项</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● True ● False <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_store_multi_tenant
swift_store_user = None	字符串值	<p>对 Swift 身份验证服务进行身份验证的用户。</p>
swift_upload_buffer_dir = None	字符串值	<p>上传到 Swift 前用于缓冲镜像片段的目录。</p> <p>提供一个字符串值，代表 glance 节点上目录的绝对路径，在镜像片段上传到 swift 之前会短暂缓冲镜像段。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 这只有在配置选项 swift_buffer_on_upload 设置为 True 时才需要。 ● 应置备此目录，请记住 swift_store_large_object_chunk_size 以及给定 glance 节点可以同时上传的最大镜像数。 <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表绝对路径的字符串值 <p>相关选项：</p> <ul style="list-style-type: none"> ● swift_buffer_on_upload ● swift_store_large_object_chunk_size

配置选项 = 默认值	类型	Description
vmware_api_retry_count = 10	整数值	<p>VMware API 重试次数。</p> <p>此配置选项指定连接相关问题或服务器 API 调用过载时必须重试 VMware ESX/VC 服务器 API 的次数。无法指定 <i>retry forever</i>。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> None
vmware_ca_file = None	字符串值	<p>CA 捆绑包文件的绝对路径。</p> <p>此配置选项可让操作员使用自定义 Certificate 授权机构文件来验证 ESX/vCenter 证书。</p> <p>如果设置了这个选项，则忽略 "vmware_insecure" 选项，指定的 CA 文件将用于验证 ESX/vCenter 服务器证书，并建立与服务器的安全连接。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何是 CA 文件的有效绝对路径的字符串 <p>相关选项：</p> <ul style="list-style-type: none"> vmware_insecure
vmware_datastores = None	多值	<p>用于存储镜像的数据存储。</p> <p>此配置选项指定镜像存储在 VMWare 存储后端中的数据存储。这个选项可多次指定，以指定多个数据存储。数据存储名称应在其数据中心路径后指定，用 ":" 分隔。可以在数据存储名称后赋予可选权重，再次通过 ":" 分隔来指定优先级。因此，所需格式将变为 <code><datacenter_path>:<datastore_name>:<optional_weight></code>。</p> <p>在添加镜像时，会选择权重最高的数据存储，除非已知镜像大小时没有足够的可用空间。如果未指定权重，则假定为 0，目录将被视为最后选择。如果多个数据存储具有相同的权重，则会选择一个带有最多可用空间的权重。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何格式的字符串：<code><datacenter_path>:<datastore_name>:<optional_weight></code> <p>相关选项：<code>* None</code></p>

配置选项 = 默认值	类型	Description
vmware_insecure = False	布尔值	<p>设置 ESX/vCenter 服务器证书验证。</p> <p>此配置选项使用布尔值来确定是否验证 ESX/vCenter 服务器证书。如果此选项设为 True，则不会验证 ESX/vCenter 服务器证书。如果此选项设置为 False，则使用默认 CA 信任存储进行验证。</p> <p>如果设置了 "vmware_ca_file" 选项，则忽略这个选项。在这种情况下，将使用使用 "vmware_ca_file" 选项指定的文件来验证 ESX/vCenter 服务器证书。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • True • False <p>相关选项：</p> <ul style="list-style-type: none"> • vmware_ca_file
vmware_server_host = None	主机地址值	<p>ESX/ESXi 或 vCenter Server 目标系统的地址。</p> <p>此配置选项设置 ESX/ESXi 或 vCenter Server 目标系统的地址。使用 VMware 存储后端时需要这个选项。地址可以包含 IP 地址(127.0.0.1)或 DNS 名称 (www.my-domain.com)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 有效的 IPv4 或 IPv6 地址 • 有效的 DNS 名称 <p>相关选项：</p> <ul style="list-style-type: none"> • vmware_server_username • vmware_server_password

配置选项 = 默认值	类型	Description
vmware_server_password = None	字符串值	<p>服务器密码。</p> <p>此配置选项使用密码通过 VMware ESX/ESXi 或 vCenter 服务器进行身份验证。使用 VMware 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串，与使用 "vmware_server_username" 选项指定的用户名对应的密码 <p>相关选项：</p> <ul style="list-style-type: none"> vmware_server_host vmware_server_username
vmware_server_username = None	字符串值	<p>服务器用户名。</p> <p>此配置选项使用用户名来通过 VMware ESX/ESXi 或 vCenter 服务器进行身份验证。使用 VMware 存储后端时需要这个选项。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串，这是具有适当权限的用户的用户名 <p>相关选项：</p> <ul style="list-style-type: none"> vmware_server_host vmware_server_password
vmware_store_image_dir = /openstack_glance	字符串值	<p>glance 镜像要存储在数据存储中的目录。</p> <p>此配置选项指定 glance 镜像存储在 VMware 数据存储中的目录的路径。如果没有设置这个选项，则存储 glance 镜像的默认目录为 openstack_glance。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何是到目录的有效路径的字符串 <p>相关选项：</p> <ul style="list-style-type: none"> None

配置选项 = 默认值	类型	Description
vmware_task_poll_interval = 5	整数值	<p>用于轮询在 VMware ESX/VC 服务器上调用的远程任务的间隔（以秒为单位）。</p> <p>此配置选项需要休眠时间（以秒为单位）轮询 on-going async 任务，作为 VMWare ESX/VC 服务器 API 调用的一部分。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数值 <p>相关选项：</p> <ul style="list-style-type: none"> None

3.3.3. oslo_policy

下表概述了 `/etc/glance/glance-cache.conf` 文件中的 `[oslo_policy]` 组下可用的选项。

表 3.31. oslo_policy

配置选项 = 默认值	类型	描述
enforce_new_defaults = False	布尔值	这个选项控制在评估策略时是否使用旧的弃用默认值。如果为 True ，则不会评估旧的已弃用的默认值。这意味着，如果旧默认值允许任何现有令牌，但不允许使用新默认值，它将被禁止。我们鼓励启用此标志以及 enforce_scope 标志，以便您可以同时获得新默认值和 scope_type 的好处
enforce_scope = False	布尔值	此选项控制在评估策略时是否强制执行范围。如果为 True ，则请求中使用的令牌范围将与被强制执行策略的 scope_types 进行比较。如果范围不匹配，则引发 InvalidScope 异常。如果为 False ，则会记录一条信息，告知操作员使用了不匹配的范围调用策略。
policy_default_rule = default	字符串值	默认规则。未找到请求的规则时强制执行。
policy_dirs = ['policy.d']	多值	存储策略配置文件的目录。它们可以相对于 <code>config_dir</code> 选项或绝对路径定义的搜索路径中的任何目录。由 <code>policy_file</code> 定义的文件必须存在，才能搜索这些目录。缺少或空目录将被忽略。
policy_file = policy.yaml	字符串值	将角色映射到给定服务权限的文件的相对或绝对路径。必须使用与配置文件设置相关的相对路径。

配置选项 = 默认值	类型	描述
remote_content_type = application/x-www-form-urlencoded	字符串值	用于发送和接收基于 REST 策略检查的数据的内容类型
remote_ssl_ca_cert_file = None	字符串值	基于 REST 的策略检查的 ca cert 文件的绝对路径
remote_ssl_client_cert_file = None	字符串值	基于 REST 的策略检查的客户端证书的绝对路径
remote_ssl_client_key_file = None	字符串值	绝对路径客户端密钥文件 REST 基于策略检查
remote_ssl_verify_server_cert = False	布尔值	基于 REST 的策略检查的服务器身份验证

第 4 章 HEAT

下面的章节包含有关 **heat** 服务中的配置选项的信息。

4.1. HEAT.CONF

本节包含 `/etc/heat/heat.conf` 文件的选项。

4.1.1. DEFAULT

下表概述了 `/etc/heat/heat.conf` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
action_retry_limit = 5	整数值	重试使资源进入非错误状态的次数。设置为 0 可禁用重试。
allow_trusts_redelegation = False	布尔值	创建启用了 redelegation 的信任。这个选项仅在 reauthentication_auth_method 设置为 "trusts" 时使用。请注意，启用此选项具有安全影响，因为 Heat 创建的所有信任都将使用模拟和重新启用。只有在其他服务需要从 Heat 用于访问它们时创建信任时启用它，例如在配置为使用信任时的 Aodh 和 Heat。
auth_encryption_key = notgood but just long enough i t	字符串值	用于加密数据库中身份验证信息的密钥。这个密钥的长度必须为 32 个字符。
backdoor_port = None	字符串值	启用事件程序后端。可接受的值为 0、<port> 和 <start>:<end>，其中 0 会导致侦听随机的 tcp 端口号；<port> 会导致监听指定的端口号（如果正在使用该端口，则不会启用 backdoor）；<start>:<end> 会在指定的端口号范围内侦听最小未使用的端口号。所选端口显示在服务的日志文件中。
backdoor_socket = None	字符串值	启用 eventlet 后端，使用提供的路径作为可接收连接的 unix 套接字。这个选项与中的 <i>backdoor_port</i> 相互排斥，该选项只应提供一个。如果提供这两个选项，则此选项的存在会覆盖该选项的使用。在路径 {pid} 中将被当前进程的 PID 替换。
client_retry_limit = 2	整数值	当客户端遇到预期的间歇性错误时，重试的次数。设置为 0 可禁用重试。
cloud_backend = heat.engine.clients.OpenStackClients	字符串值	用作客户端后端的完全限定类名称。
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制

配置选项 = 默认值	类型	Description
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
control_exchange = openstack	字符串值	默认交换，哪些主题在哪些方面有范围。可以被 transport_url 选项指定的交换名称覆盖。
convergence_engine = True	布尔值	启用带有聚合架构的引擎。所有带有此选项的堆栈都将使用聚合引擎创建。
debug = False	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。
default_deployment_signal_transport = CFN_SIGNAL	字符串值	服务器应如何向 heat 信号显示部署输出值的模板默认值。CFN_SIGNAL 将允许 HTTP POST 到 CFN 密钥对签名 URL（需要启用 heat-api-cfn）。TEMP_URL_SIGNAL 将创建一个 Swift TempURL，以通过 HTTP PUT 发出信号（需要支持 TempURL 的 object-store 端点）。HEAT_SIGNAL 将允许使用提供的 keystone 凭据调用 Heat API 资源信号。ZAQAR_SIGNAL 将创建一个专用的 zaqar 队列，以使用提供的 keystone 凭证发出信号。
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib3.connectionpool=WARN', 'urllib3.connectionpool=WARN', 'websocket=WARN', 'requests.packages.urllib3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WARN', 'routes.middleware=WARN', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INFO']	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 log_config_append，则忽略此选项。

配置选项 = 默认值	类型	Description
default_notification_level = INFO	字符串值	传出通知的默认通知级别。
default_publisher_id = None	字符串值	用于传出通知的默认 publisher_id。
default_software_config_transport = POLL_SERVER_CFN	字符串值	服务器应如何接收软件配置所需的元数据的模板默认。POLL_SERVER_CFN 将允许调用通过提供的密钥对进行身份验证的 cfn API 操作 DescribeStackResource（需要启用 heat-api-cfn）。POLL_SERVER_HEAT 将允许使用提供的 keystone 凭证调用 Heat API resource-show（需要 keystone v3 API 和配置的 stack_userptr 配置选项）。POLL_TEMP_URL 将创建并填充用于轮询的元数据的 Swift TempURL（需要支持 TempURL 的 object-store 端点）。ZAQAR_MESSAGE 将创建一个专用的 zaqar 队列并发布用于轮询的元数据。
default_user_data_format = HEAT_CFNTOOLS	字符串值	模板默认值，用于如何为服务器格式化 user_data。对于 HEAT_CFNTOOLS，user_data 作为 heat-cfntools cloud-init 引导配置数据的一部分被捆绑。对于 RAW，user_data 被不修改地传递到 Nova。对于 SOFTWARE_CONFIG user_data 作为软件配置数据的一部分，元数据则派生自任何关联的 SoftwareDeployment 资源。
deferred_auth_method = trusts	字符串值	选择 deferred auth 方法，存储了密码或信任。 弃用自：9.0.0 在与 keystone v3 一起使用且不被支持时，基于 reason:*Stored 密码的延迟 auth 无法正常工作。
enable_cloud_watch_lite = False	布尔值	启用传统的 OS::Heat::CWLiteAlarm 资源。 弃用自：10.0.0 *reason:*Heat CloudWatch Service 已被删除。
enable_stack_abandon = False	布尔值	启用 preview Stack Abandon 功能。
enable_stack_adopt = False	布尔值	启用 preview Stack Adopt 功能。
encrypt_parameters_and_properties = False	布尔值	加密标记为隐藏的模板参数，以及将所有资源属性存储在数据库之前。
engine_life_check_timeout = 2	整数值	用于堆栈锁定的引擎存活度检查的 RPC 超时。

配置选项 = 默认值	类型	Description
environment_dir = /etc/heat/environment.d	字符串值	用于搜索环境文件的目录。
error_wait_time = 240	整数值	发生错误后可以继续运行任务在被取消前可以继续运行的时间（以秒为单位）。
event_purge_batch_size = 200	整数值	控制在清除堆栈的事件时，将修剪多少个事件。将其设置为更频繁的清除时保留更多事件。
executor_thread_pool_size = 64	整数值	当 executor 为 threading 或 eventlet 时，executor 线程池的大小。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
graceful_shutdown_timeout = 60	整数值	指定安全关闭服务器将退出的超时。零值表示无限期等待。
heat_metadata_server_url = None	字符串值	Heat 元数据服务器的 URL。注意：只有在您需要实例使用与 keystone 目录中不同的端点时，才需要设置此设置
heat_stack_user_role = heat_stack_user	字符串值	heat 模板定义用户的 Keystone 角色。
heat_waitcondition_server_url = None	字符串值	Heat waitcondition 服务器的 URL。
<code>heat_watch_server_url = `</code>	字符串值	Heat CloudWatch 服务器的 URL。 弃用自：10.0.0 *reason:*Heat CloudWatch Service 已被删除。
hidden_stack_tags = ['data-processing-cluster']	列表值	包含这些标签名称的堆栈将被隐藏。应以逗号分隔的列表指定多个标签（例如，hidden_stack_tags=hide_me,me_too）。
Host = <based on operating system>	字符串值	引擎节点的名称。这可以是不透明标识符。它不一定是主机名、FQDN 或 IP 地址。
instance_connection_https_validate_certificates = 1	字符串值	如果使用 SSL，与 CFN/CW API 实例连接会验证证书。
instance_connection_is_secure = 0	字符串值	通过 https 连接到 CFN/CW API 实例。

配置选项 = 默认值	类型	Description
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。
keystone_backend = heat.engine.clients.os.keystone.heat_keystoneclient.KsClientWrapper	字符串值	用作 keystone 后端的完全限定类名称。
loadbalancer_template = None	字符串值	内置 loadbalancer 嵌套堆栈的自定义模板。
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 <code>log-date-format</code> ）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_options = True	布尔值	启动服务时，启用或禁用所有注册选项的日志值（在 <code>DEBUG</code> 级别）。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。

配置选项 = 默认值	类型	Description
logging_context_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 DEBUG 时，要附加到日志消息的其他数据。 oslo_log.formatters.ContextFormatter 使用
logging_default_format_s tring = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] % (instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_for mat = %(user) s %(tenant) s s %(domain) s % (user_domain) s % (project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_events_per_stack = 1000	整数值	粗略每个堆栈可用的最大事件数。实际事件数量可能会更高，因为清除检查会花费随机 200/event_purge_batch_size 百分比。当事件被清除时，会删除旧的事件。为每个堆栈限制事件设置为 0。
max_interface_check_att empts = 10	整数值	检查接口是否已附加或分离的次数。
max_ironic_api_microver sion = None	浮点值	客户端插件的最大 ironic API 版本。在这个版本中，任何支持 max_ironic_api_microversion 以上的微版本号支持的任何 ironic 功能都不可用。
max_json_body_size = 1048576	整数值	JSON 请求正文的最大原始字节大小。应该大于 max_template_size。

配置选项 = 默认值	类型	Description
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
max_nested_stack_depth = 5	整数值	使用嵌套堆栈时允许的最大深度。
max_nova_api_microversion = None	浮点值	客户端插件的最大 nova API 版本。在这个版本中，任何支持 max_nova_api_microversion 以上的微版本号支持的 nova 功能都不可用。
max_resources_per_stack = 1000	整数值	每个顶级堆栈允许的最大资源数。-1 代表无限量。
max_server_name_length = 53	整数值	在 nova 中使用的服务器名称的最大长度。
max_stacks_per_tenant = 512	整数值	一个租户一次可能具有的最大堆栈数。-1 代表无限。
max_template_size = 524288	整数值	任何模板的最大原始字节大小。
num_engine_workers = None	整数值	用于分叉和运行的 heat-engine 进程数量。将默认为主机上的 4 或 CPU 数量，以较大者为准。
observe_on_update = False	布尔值	在更新时，允许 heat 从事实上收集现有资源属性，并聚合到更新的模板。
onready = None	字符串值	已弃用。
periodic_interval = 60	整数值	运行定期任务之间的秒。
plugin_dirs = ['/usr/lib64/heat', '/usr/lib/heat', '/usr/local/lib/heat', '/usr/local/lib64/heat']	列表值	用于搜索插件的目录列表。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。

配置选项 = 默认值	类型	Description
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
<code>`reauthentication_auth_method = `</code>	字符串值	允许对令牌到期进行重新身份验证，以便长时间运行的任务可能完成。请注意，这会破坏任何提供用户令牌的到期。
region_name_for_services = None	字符串值	用于获取服务端点的默认区域名称。
region_name_for_shared_services = None	字符串值	共享服务端点的区域名称。
rpc_conn_pool_size = 30	整数值	RPC 连接池的大小。
rpc_ping_enabled = False	布尔值	添加端点以回答 ping 调用。端点名为 oslo_rpc_server_ping
rpc_response_timeout = 60	整数值	从调用等待响应的秒数。
run_external_periodic_tasks = True	布尔值	某些定期任务可以在单独的进程中运行。我们是否应该在此处运行？
<code>`server_keystone_endpoint_type = `</code>	字符串值	如果设置，则用于控制用户控制的服务器使用哪些身份验证端点来重新调用 Heat。如果使用未设置 www_authenticate_uri。
shared_services_types = ['image', 'volume', 'volumev2']	列表值	位于其他地区中的共享服务。Needs region_name_for_shared_services 选项设为此效果。
stack_action_timeout = 3600	整数值	堆栈操作的超时时间（如 create 或 update）。
stack_domain_admin = None	字符串值	Keystone 用户名，具有角色的用户足以管理 stack_user_domain 中的用户和项目。
stack_domain_admin_password = None	字符串值	stack_domain_admin 用户的 Keystone 密码。

配置选项 = 默认值	类型	Description
stack_scheduler_hints = False	布尔值	启用此功能后，标识服务器或卷资源的 heat 堆栈上下文的调度程序提示将传递到 nova 和 cinder 中的已配置调度程序，例如使用 heat 资源类型 OS::Cinder::Volume、OS::Nova::Server 和 AWS::EC2::Instance。heat_root_stack_id 将设置为资源的根堆栈的 id，heat_stack_id 将设置为资源父堆栈的 id，heat_stack_name 将设置为资源父堆栈的名称，heat_path_in_stack 将设置为以逗号分隔的 stackresource name 和带有 list[0] 的 stackname 的列表，heat_resource_name 将设为资源名称，即 heat_resource_name。和 heat_resource_uuid 将被设为资源的编配 ID。
stack_user_domain_id = None	字符串值	包含 heat 模板定义用户的 Keystone 域 ID。如果设置了这个选项，则忽略 stack_user_domain_name 选项。
stack_user_domain_name = None	字符串值	包含 heat 模板定义用户的 Keystone 域名。如果设置了 stack_user_domain_id 选项，则忽略此选项。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
template_dir = /etc/heat/templates	字符串值	用于搜索模板文件的目录。
transport_url = rabbit://	字符串值	用于连接到消息传递后端的网络地址和可选用户凭证，采用 URL 格式。预期格式为： driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query 示例：rabbit://rabbitmq:password@127.0.0.1:5672// 有关 URL 中字段的详情，请查看 oslo_messaging.TransportURL 文档 https://docs.openstack.org/oslo.messaging/latest/reference/transport.html
trusts_delegated_roles = []	列表值	要委派给 heat 的信任者角色的子集。如果未设置，则在创建堆栈时，用户的所有角色都会被委派给 heat。
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。

配置选项 = 默认值	类型	Description
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 log_file 选项并使用 Linux 平台时，才有意义。如果设置了 log_config_append，则忽略此选项。

4.1.2. auth_password

下表概述了 /etc/heat/heat.conf 文件中的 [auth_password] 组下可用的选项。

表 4.1. auth_password

配置选项 = 默认值	类型	Description
allowed_auth_uris = []	列表值	启用 multi_cloud 时，auth_uri 允许的 keystone 端点。至少需要指定一个端点。
multi_cloud = False	布尔值	允许编排多个云。

4.1.3. 客户端

下表概述了 /etc/heat/heat.conf 文件中的 [clients] 组下可用的选项。

表 4.2. 客户端

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = publicURL	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = False	布尔值	如果设置，则不会验证服务器证书。

配置选项 = 默认值	类型	Description
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.4. clients_aodh

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_aodh]` 组下可用的选项。

表 4.3. clients_aodh

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.5. clients_barbican

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_barbican]` 组下可用的选项。

表 4.4. clients_barbican

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.6. clients_cinder

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_cinder]` 组下可用的选项。

表 4.5. clients_cinder

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
http_log_debug = False	布尔值	允许客户端的调试日志输出。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.7. clients_designate

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_designate]` 组下可用的选项。

表 4.6. clients_designate

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.8. clients_glance

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_glance]` 组下可用的选项。

表 4.7. clients_glance

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。

配置选项 = 默认值	类型	Description
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.9. clients_heat

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_heat]` 组下可用的选项。

表 4.8. clients_heat

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。
<code>`url = `</code>	字符串值	可选的 heat url 格式，如 http://0.0.0.0:8004/v1/(tenant_id)s 。

4.1.10. clients_keystone

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_keystone]` 组下可用的选项。

表 4.9. clients_keystone

配置选项 = 默认值	类型	Description
<code>`auth_uri = `</code>	字符串值	未指定版本的 keystone url 格式，如 http://0.0.0.0:5000 。
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。

配置选项 = 默认值	类型	Description
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.11. clients_magnum

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_magnum]` 组下可用的选项。

表 4.10. clients_magnum

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.12. clients_manila

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_manila]` 组下可用的选项。

表 4.11. clients_manila

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.13. clients_mistral

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_mistral]` 组下可用的选项。

表 4.12. clients_mistral

配置选项 = 默认值	类型	Description
<code>ca_file = None</code>	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
<code>cert_file = None</code>	字符串值	可选 PEM 格式的证书链文件。
<code>endpoint_type = None</code>	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
<code>insecure = None</code>	布尔值	如果设置，则不会验证服务器证书。
<code>key_file = None</code>	字符串值	包含私钥的可选 PEM 格式文件。

4.1.14. clients_monasca

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_monasca]` 组下可用的选项。

表 4.13. clients_monasca

配置选项 = 默认值	类型	Description
<code>ca_file = None</code>	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
<code>cert_file = None</code>	字符串值	可选 PEM 格式的证书链文件。
<code>endpoint_type = None</code>	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
<code>insecure = None</code>	布尔值	如果设置，则不会验证服务器证书。
<code>key_file = None</code>	字符串值	包含私钥的可选 PEM 格式文件。

4.1.15. clients_neutron

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_neutron]` 组下可用的选项。

表 4.14. clients_neutron

配置选项 = 默认值	类型	Description
<code>ca_file = None</code>	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
<code>cert_file = None</code>	字符串值	可选 PEM 格式的证书链文件。

配置选项 = 默认值	类型	Description
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.16. clients_nova

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_nova]` 组下可用的选项。

表 4.15. clients_nova

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
http_log_debug = False	布尔值	允许客户端的调试日志输出。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.17. clients_octavia

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_octavia]` 组下可用的选项。

表 4.16. clients_octavia

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。

配置选项 = 默认值	类型	Description
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.18. clients_sahara

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_sahara]` 组下可用的选项。

表 4.17. clients_sahara

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.19. clients_senlin

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_senlin]` 组下可用的选项。

表 4.18. clients_senlin

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.20. clients_swift

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_swift]` 组下可用的选项。

表 4.19. clients_swift

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.21. clients_trove

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_trove]` 组下可用的选项。

表 4.20. clients_trove

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.22. clients_vitrage

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_vitrage]` 组下可用的选项。

表 4.21. clients_vitrage

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。

配置选项 = 默认值	类型	Description
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.23. clients_zaqar

下表概述了 `/etc/heat/heat.conf` 文件中的 `[clients_zaqar]` 组下可用的选项。

表 4.22. clients_zaqar

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。
endpoint_type = None	字符串值	身份服务目录中的端点类型，用于与 OpenStack 服务通信。
insecure = None	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。

4.1.24. CORS

下表概述了 `/etc/heat/heat.conf` 文件中的 `[cors]` 组下可用的选项。

表 4.23. CORS

配置选项 = 默认值	类型	Description
allow_credentials = True	布尔值	指明实际请求可以包含用户凭证
allow_headers = ['X-Auth-Token', 'X-Identity-Status', 'X-Roles', 'X-Service-Catalog', 'X-User-Id', 'X-Tenant-Id', 'X-OpenStack-Request-ID']	列表值	指定在实际请求过程中可以使用哪些标头字段名称。
allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH']	列表值	指明在实际请求期间可以使用哪些方法。
allowed_origin = None	列表值	指明此资源是否可以与请求"origin"标头中收到的域共享。格式： <code>"<protocol>://<host>[:<port>]"</code> ，没有尾部斜杠。示例： https://horizon.example.com

配置选项 = 默认值	类型	Description
expose_headers = ['X-Auth-Token', 'X-Subject-Token', 'X-Service-Token', 'X-OpenStack-Request-ID']	列表值	指明哪些标头可以安全地公开给 API。默认为 HTTP Simple Headers。
max_age = 3600	整数值	CORS preflight 请求的最大缓存期限。

4.1.25. database

下表概述了 `/etc/heat/heat.conf` 文件中的 `[database]` 组下可用的选项。

表 4.24. database

配置选项 = 默认值	类型	Description
backend = sqlalchemy	字符串值	用于数据库的后端。
connection = None	字符串值	用于连接到数据库的 SQLAlchemy 连接字符串。
connection_debug = 0	整数值	SQL 调试信息的详细程度：0=None, 100=Everything。
<code>`connection_parameters = `</code>	字符串值	用于连接时附加到连接 URL 的可选 URL 参数；指定 <code>param1=value1&param2=value2&...</code>
connection_recycle_time = 3600	整数值	连接池中存在的连接的时间超过这个秒数，下一次从池中签出它们的时间将替换为新的连接。
connection_trace = False	布尔值	将 Python 堆栈跟踪作为注释字符串添加到 SQL。
db_inc_retry_interval = True	布尔值	如果为 True，将数据库操作的重试间隔增加到 <code>db_max_retry_interval</code> 。
db_max_retries = 20	整数值	在引发错误前出现连接错误或死锁错误时的最大重试次数。设置为 -1 以指定无限重试计数。
db_max_retry_interval = 10	整数值	如果设置了 <code>db_inc_retry_interval</code> ，则数据库操作重试之间的最大秒数。
db_retry_interval = 1	整数值	数据库事务重试之间的秒数。

配置选项 = 默认值	类型	Description
max_overflow = 50	整数值	如果设置，请将这个值用于 SQLAlchemy 的 max_overflow。
max_pool_size = 5	整数值	在池中保留的最大 SQL 连接数。将值设为 0 表示没有限制。
max_retries = 10	整数值	启动期间数据库连接重试的最大数量。设置为 -1 以指定无限重试计数。
mysql_enable_ndb = False	布尔值	如果为 True，则透明地启用对处理 MySQL 集群 (NDB) 的支持。
mysql_sql_mode = TRADITIONAL	字符串值	用于 MySQL 会话的 SQL 模式。这个选项（包括默认选项）会覆盖任何 server-set SQL 模式。要使用服务器配置设置的任何 SQL 模式，请将其设置为 no 值。示例：mysql_sql_mode=
pool_timeout = None	整数值	如果设置，请将这个值用于 SQLAlchemy 的 pool_timeout。
retry_interval = 10	整数值	重新尝试打开 SQL 连接之间的间隔。
slave_connection = None	字符串值	用于连接到从数据库的 SQLAlchemy 连接字符串。
sqlite_synchronous = True	布尔值	如果为 True，则 SQLite 使用同步模式。
use_db_reconnect = False	布尔值	启用在连接丢失时进行数据库重新连接的实验性使用。

4.1.26. ec2authtoken

下表概述了 `/etc/heat/heat.conf` 文件中的 `[ec2authtoken]` 组下可用的选项。

表 4.25. ec2authtoken

配置选项 = 默认值	类型	Description
allowed_auth_uris = []	列表值	启用 multi_cloud 时，auth_uri 允许的 keystone 端点。至少需要指定一个端点。
auth_uri = None	字符串值	身份验证端点 URI。
ca_file = None	字符串值	在 SSL 连接中使用的可选 CA 证书文件。
cert_file = None	字符串值	可选 PEM 格式的证书链文件。

配置选项 = 默认值	类型	Description
insecure = False	布尔值	如果设置，则不会验证服务器证书。
key_file = None	字符串值	包含私钥的可选 PEM 格式文件。
multi_cloud = False	布尔值	允许编排多个云。

4.1.27. eventlet_opts

下表概述了 `/etc/heat/heat.conf` 文件中的 `[eventlet_opts]` 组下可用的选项。

表 4.26. eventlet_opts

配置选项 = 默认值	类型	Description
client_socket_timeout = 900	整数值	客户端连接的套接字操作的超时。如果传入的连接闲置了这个秒数，它将被关闭。值 0 表示等待永久。
wsgi_keep_alive = True	布尔值	如果为 False，请明确关闭客户端套接字连接。

4.1.28. healthcheck

下表概述了 `/etc/heat/heat.conf` 文件中的 `[healthcheck]` 组下可用的选项。

表 4.27. healthcheck

配置选项 = 默认值	类型	Description
backends = []	列表值	可执行健康检查并报告该信息作为请求的一部分的其他后端。
detailed = False	布尔值	显示响应中更详细的信息。安全备注：启用此选项可能会公开有关被监控服务的敏感详情。务必验证它是否不会违反您的安全策略。
disable_by_file_path = None	字符串值	检查是否存在文件，以确定应用是否在端口上运行。DisableByFileHealthcheck 插件使用。
disable_by_file_paths = []	列表值	根据端口检查文件是否存在，以确定应用是否在端口上运行。需要字符串的 "port:path" 列表。DisableByFilesPortsHealthcheck 插件使用。
path = /healthcheck	字符串值	响应 healthcheck 请求的路径。

4.1.29. heat_api

下表概述了 `/etc/heat/heat.conf` 文件中的 `[heat_api]` 组下可用的选项。

表 4.28. heat_api

配置选项 = 默认值	类型	Description
backlog = 4096	整数值	要配置套接字的积压请求数。
bind_host = 0.0.0.0	IP 地址值	绑定服务器的地址。在选择特定网络接口时很有用。
bind_port = 8004	端口值	服务器将侦听的端口。
cert_file = None	字符串值	用于 SSL 模式的 SSL 证书文件的位置。
key_file = None	字符串值	用于启用 SSL 模式的 SSL 密钥文件的位置。
max_header_line = 16384	整数值	要接受的最大消息标头大小。在使用大型令牌时（通常是由 Keystone v3 API 为大型服务目录生成的），可能需要增加 max_header_line。
tcp_keepidle = 600	整数值	套接字选项 TCP_KEEPIDLE 的值。这是在 TCP 开始发送 keepalive 探测前必须闲置连接的时间（以秒为单位）。
workers = 0	整数值	Heat 服务的 worker 数量。默认值 0 表示，该服务将启动与服务器上内核相等的 worker 数量。

4.1.30. heat_api_cfn

下表概述了 `/etc/heat/heat.conf` 文件中的 `[heat_api_cfn]` 组下可用的选项。

表 4.29. heat_api_cfn

配置选项 = 默认值	类型	Description
backlog = 4096	整数值	要配置套接字的积压请求数。
bind_host = 0.0.0.0	IP 地址值	绑定服务器的地址。在选择特定网络接口时很有用。
bind_port = 8000	端口值	服务器将侦听的端口。
cert_file = None	字符串值	用于 SSL 模式的 SSL 证书文件的位置。
key_file = None	字符串值	用于启用 SSL 模式的 SSL 密钥文件的位置。
max_header_line = 16384	整数值	要接受的最大消息标头大小。在使用大型令牌时（通常是由 Keystone v3 API 为大型服务目录生成的），可能需要增加 max_header_line。

配置选项 = 默认值	类型	Description
tcp_keepidle = 600	整数值	套接字选项 TCP_KEEPIDLE 的值。这是在 TCP 开始发送 keepalive 探测前必须闲置连接的时间（以秒为单位）。
workers = 1	整数值	Heat 服务的 worker 数量。

4.1.31. heat_api_cloudwatch

下表概述了 `/etc/heat/heat.conf` 文件中的 `[heat_api_cloudwatch]` 组下可用的选项。

表 4.30. heat_api_cloudwatch

配置选项 = 默认值	类型	Description
backlog = 4096	整数值	要配置套接字的积压请求数。弃用自：10.0.0 *reason:*Heat CloudWatch API 已被删除。
bind_host = 0.0.0.0	IP 地址值	绑定服务器的地址。在选择特定网络接口时很有用。弃用自：10.0.0 *reason:*Heat CloudWatch API 已被删除。
bind_port = 8003	端口值	服务器将侦听的端口。弃用自：10.0.0 *reason:*Heat CloudWatch API 已被删除。
cert_file = None	字符串值	用于 SSL 模式的 SSL 证书文件的位置。弃用自：10.0.0 *reason:*Heat CloudWatch API 已被删除。
key_file = None	字符串值	用于启用 SSL 模式的 SSL 密钥文件的位置。弃用自：10.0.0 *reason:*Heat CloudWatch API 已被删除。
max_header_line = 16384	整数值	要接受的最大消息标头大小。在使用大型令牌时（通常是由 Keystone v3 API 使用大型服务目录生成的），则可能需要增加 <code>max_header_line</code> 弃用自：10.0.0 *reason:*Heat CloudWatch API 已被删除。
tcp_keepidle = 600	整数值	套接字选项 TCP_KEEPIDLE 的值。这是在 TCP 开始发送 keepalive 探测前必须闲置连接的时间（以秒为单位）。弃用自：10.0.0 *reason:*Heat CloudWatch API 已被删除。

配置选项 = 默认值	类型	Description
workers = 1	整数值	Heat 服务的 worker 数量。弃用自：10.0.0 *reason:*Heat CloudWatch API 已被删除。

4.1.32. keystone_authtoken

下表概述了 `/etc/heat/heat.conf` 文件中的 `[keystone_authtoken]` 组下可用的选项。

表 4.31. keystone_authtoken

配置选项 = 默认值	类型	Description
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
auth_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。这个选项已弃用，现在使用 <code>www_authenticate_uri</code> ，并将在 S 发行版本中删除。弃用自：Queens *reason:* auth_uri 选项已弃用，现在使用 <code>www_authenticate_uri</code> ，并将在 S 发行版本中删除。
auth_version = None	字符串值	Identity API 端点的 API 版本。
cache = None	字符串值	存储 Swift 缓存对象的请求环境密钥。当使用 Swift 缓存部署 <code>auth_token</code> 中间件时，请使用此选项使中间件共享具有 swift 的缓存后端。否则，使用 <code>memcached_servers</code> 选项。
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。默认为系统 CA。
certFile = None	字符串值	如果身份服务器需要客户端证书，则需要此项
delay_auth_decision = False	布尔值	不要在中间件内处理授权请求，而是将授权决策委派给下游 WSGI 组件。

配置选项 = 默认值	类型	Description
enforce_token_bind = permissive	字符串值	用于控制令牌绑定的使用和类型。可以设置为 "disabled" 以不检查令牌绑定。如果绑定类型是已知服务器的绑定类型，则验证绑定信息，并忽略它（如 "permissive"），但如果绑定类型未知，则令牌将被拒绝。"必需"令牌绑定需要允许任何形式的令牌绑定。最后，令牌中必须存在的绑定方法名称。
http_connect_timeout = None	整数值	请求超时值与身份 API 服务器通信。
http_request_max_retries = 3	整数值	与身份 API 服务器通信时，我们尝试重新连接的次数。
include_service_catalog = True	布尔值	（可选）指示是否设置 X-Service-Catalog 标头。如果为 False，则中间件不会要求在令牌验证时请求服务目录，且不会设置 X-Service-Catalog 标头。
insecure = False	布尔值	验证 HTTPS 连接。
interface = internal	字符串值	用于 Identity API 端点的接口。有效值为 "public", "internal"（默认）或 "admin"。
keyfile = None	字符串值	如果身份服务器需要客户端证书，则需要此项
memcache_pool_conn_get_timeout = 10	整数值	（可选）操作等待从池中获取 memcached 客户端连接的秒数。
memcache_pool_dead_retry = 300	整数值	（可选）在再次尝试前 memcached 服务器被视为 dead 的秒数。
memcache_pool_maxsize = 10	整数值	（可选）到每个 memcached 服务器的最大打开连接数。
memcache_pool_socket_timeout = 3	整数值	（可选）与 memcached 服务器通信的套接字超时（以秒为单位）。
memcache_pool_unused_timeout = 60	整数值	（可选）与 memcached 的连接在池关闭前未使用的秒数。
memcache_secret_key = None	字符串值	（可选，如果定义了 memcache_security_strategy）此字符串用于密钥派生。

配置选项 = 默认值	类型	Description
memcache_security_strategy = None	字符串值	(可选) 如果定义, 则指示令牌数据是否应进行身份验证或加密。如果 MAC, 令牌数据在缓存中进行身份验证 (使用 HMAC)。如果 ENCRYPT, 令牌数据会在缓存中加密并进行身份验证。如果该值不是这些选项之一或为空, 则 auth_token 会在初始化时引发异常。
memcache_use_advanced_pool = False	布尔值	(可选) 使用 advanced (事件安全) memcached 客户端池。高级池只能在 python 2.x 下工作。
memcached_servers = None	列表值	(可选) 指定要用于缓存的 memcached 服务器列表。如果未定义, 令牌将会被缓存在进程中。
region_name = None	字符串值	找到身份服务器的区域。
service_token_roles = ['service']	列表值	需要出现在服务令牌中的角色。服务令牌可以请求可以使用过期的令牌, 因此此检查应严格控制只有实际服务应发送此令牌。此处的角色作为 ANY 检查应用, 因此此列表中的任何角色都必须存在。为了向后兼容的原因, 这目前只会影响 allow_expired 检查。
service_token_roles_required = False	布尔值	为了向后兼容的原因, 我们必须允许有效的服务令牌传递, 但不允许将 service_token_roles 检查视为有效。在以后的发行版本中设置此 true 将成为默认设置, 并应启用 (如果可能)。
service_type = None	字符串值	服务的名称或类型 (在服务目录中出现)。这用于验证具有受限访问规则的令牌。
token_cache_time = 300	整数值	为了防止过度努力验证令牌, 中间件会缓存之前可预见的令牌 (以秒为单位)。设置为 -1 可完全禁用缓存。
www_authenticate_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点, 因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下, 这个端点应该被未指定版本, 但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点, 则该服务用户不应该用于验证令牌的端点, 因为普通用户可能无法访问该端点。

4.1.33. noauth

下表概述了 `/etc/heat/heat.conf` 文件中的 **[noauth]** 组下可用的选项。

表 4.32. noauth

配置选项 = 默认值	类型	Description
<code>`token_response = `</code>	字符串值	包含 noauth 中间件返回内容的 JSON 文件。

4.1.34. oslo_messaging_amqp

下表概述了 `/etc/heat/heat.conf` 文件中的 `[oslo_messaging_amqp]` 组下可用的选项。

表 4.33. oslo_messaging_amqp

配置选项 = 默认值	类型	Description
addressing_mode = dynamic	字符串值	表示驱动程序使用的寻址模式。允许的值： <i>legacy</i> - 使用旧的不可路由寻址； <i>routable</i> - 使用可路由地址； <i>dynamic</i> - 如果消息总线不支持路由，使用 <i>legacy</i> ，如果支持，使用 <i>routable</i> 。
anycast_address = anycast	字符串值	在发送到一组消费者时附加到地址前缀。消息总线用来识别应在消费者之间以轮循方式发送的消息。
broadcast_prefix = broadcast	字符串值	广播到所有服务器时使用的地址前缀
connection_retry_backoff = 2	整数值	每次失败故障转移尝试后，将 <code>connection_retry_interval</code> 增加到这秒数。
connection_retry_interval = 1	整数值	尝试重新连接前暂停的秒数。
connection_retry_interval_max = 30	整数值	<code>connection_retry_interval</code> + <code>connection_retry_backoff</code> 的最大限制
container_name = None	字符串值	AMQP 容器的名称必须全局唯一。默认为生成的 UUID
default_notification_exchange = None	字符串值	通知地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_notification_exchange if set else control_exchange if set else <i>notify</i>
default_notify_timeout = 30	整数值	发送通知消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_reply_retry = 0	整数值	重新发送回复消息的最大尝试次数，因为可恢复错误而失败。
default_reply_timeout = 30	整数值	rpc 回复消息发送的截止时间。

配置选项 = 默认值	类型	Description
default_rpc_exchange = None	字符串值	RPC 地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_rpc_exchange if set other control_exchange if set else <i>rpc</i>
default_send_timeout = 30	整数值	<i>rpc cast</i> 或调用消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_sender_link_timeout = 600	整数值	调度闲置发送者链接清除的持续时间。过期后分离链接。
group_request_prefix = unicast	字符串值	发送到组中任何服务器时的地址前缀
idle_timeout = 0	整数值	不活跃连接的超时时间（以秒为单位）
link_retry_delay = 10	整数值	重新连接 AMQP 1.0 链接之间的暂停时间，该链接因为可恢复的错误而失败。
multicast_address = multicast	字符串值	在发送异常消息时，附加到地址前缀。消息总线用来识别 fanout 消息。
notify_address_prefix = openstack.org/om/notify	字符串值	所有生成的通知地址的地址前缀
notify_server_credit = 100	整数值	传入通知消息的窗口大小
pre_settled = ['rpc-cast', 'rpc-reply']	多值	发送此类型的消息预先组装。预先设置的消息不会接收来自 peer 的确认信息。请注意：如果发送失败，则预先设置的消息可能会被静默丢弃。允许的值： <i>rpc-call</i> - 发送 RPC Calls <i>pre-settledrpc-reply</i> - 发送 RPC Replies <i>pre-settled rpc-cast</i> - Send RPC Casts <i>pre-settled notify</i> - Send Notifications <i>pre-settled</i>
pseudo_vhost = True	布尔值	为不支持虚拟主机的消息总线启用虚拟主机支持（如 <i>qpidd</i> ）。当设置为 <i>true</i> 时，虚拟主机名将添加到所有消息总线地址中，从而有效地为每个虚拟主机 <i>创建专用子网</i> 。如果消息总线支持使用 AMQP 1.0 Open performative 中的 <i>hostname</i> 字段作为虚拟主机的名称，设置为 <i>False</i> 。
reply_link_credit = 200	整数值	传入 RPC 消息的窗口大小。
rpc_address_prefix = openstack.org/om/rpc	字符串值	所有生成的 RPC 地址的地址前缀
rpc_server_credit = 100	整数值	传入 RPC Request 消息的窗口大小

配置选项 = 默认值	类型	Description
<code>`sasl_config_dir = `</code>	字符串值	包含 SASL 配置的目录路径
<code>`sasl_config_name = `</code>	字符串值	配置文件名称（不带 .conf 后缀）
<code>`sasl_default_realm = `</code>	字符串值	如果用户名中没有 realm，则要使用的 SASL 域
<code>`sasl_mechanisms = `</code>	字符串值	可接受的 SASL 机制的以空格分隔的列表
server_request_prefix = exclusive	字符串值	发送到特定服务器时使用的地址前缀
ssl = False	布尔值	尝试通过 SSL 连接。如果没有给出其他与 ssl 相关的参数，它将使用系统的 CA-bundle 来验证服务器证书。
<code>`ssl_ca_file = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_cert_file = `</code>	字符串值	自我标识用于客户端身份验证的证书 PEM 文件
<code>`ssl_key_file = `</code>	字符串值	用于为 ssl_cert_file 证书签名的私钥 PEM 文件（可选）
ssl_key_password = None	字符串值	用于解密 ssl_key_file 的密码（如果加密）
ssl_verify_vhost = False	布尔值	默认情况下，SSL 检查服务器证书中的名称与 transport_url 中的主机名匹配。在某些配置中，最好使用虚拟主机名。例如，如果服务器使用 Server Name Indication TLS extension (rfc6066) 来为每个虚拟主机提供证书。如果服务器的 SSL 证书使用虚拟主机名称而不是 DNS 名称，则将 ssl_verify_vhost 设置为 True。
trace = False	布尔值	debug : 将 AMQP 帧转储到 stdout
unicast_address = unicast	字符串值	在发送到特定 RPC/Notification 服务器时，附加到地址前缀。消息总线用来识别发送到单个目的地的消息。

4.1.35. oslo_messaging_kafka

下表概述了 `/etc/heat/heat.conf` 文件中的 `[oslo_messaging_kafka]` 组下可用的选项。

表 4.34. oslo_messaging_kafka

配置选项 = 默认值	类型	Description
compression_codec = none	字符串值	producer 生成的所有数据的压缩 codec。如果没有设置，则不会使用压缩。请注意，这允许的值取决于 kafka 版本
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
consumer_group = oslo_messaging_consumer	字符串值	Kafka 消费者的组 ID。一个组中的消费者将协调消息消耗
enable_auto_commit = False	布尔值	启用异步消费者提交
kafka_consumer_timeout = 1.0	浮点值	Kafka 用户的默认超时
kafka_max_fetch_bytes = 1048576	整数值	Kafka consumer 的最大获取字节数
max_poll_records = 500	整数值	轮询调用中返回的最大记录数
pool_size = 10	整数值	Kafka Consumers 的池大小
producer_batch_size = 16384	整数值	producer async 发送的批处理大小
producer_batch_timeout = 0.0	浮点值	以秒为单位的 KafkaProducer 批处理延迟上限
sasl_mechanism = PLAIN	字符串值	当安全协议是 SASL 时的机制
security_protocol = PLAINTEXT	字符串值	用于与代理通信的协议
<code>`ssl_cafile = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_client_cert_file = `</code>	字符串值	用于身份验证的客户端证书 PEM 文件。
<code>`ssl_client_key_file = `</code>	字符串值	用于身份验证的客户端密钥 PEM 文件。
<code>`ssl_client_key_password = `</code>	字符串值	用于身份验证的客户端密钥密码文件。

4.1.36. oslo_messaging_notifications

下表概述了 `/etc/heat/heat.conf` 文件中的 `[oslo_messaging_notifications]` 组下可用的选项。

表 4.35. oslo_messaging_notifications

配置选项 = 默认值	类型	Description
driver = []	多值	处理发送通知的驱动程序。可能的值有 messaging, messagingv2, routing, log, test, noop
retry = -1	整数值	重新发送因为可恢复错误而发送失败的通知消息的最大尝试次数。0 - 不重试, -1 - indefinite
topics = ['notifications']	列表值	用于 OpenStack 通知的 AMQP 主题。
transport_url = None	字符串值	代表用于通知的消息传递驱动程序的 URL。如果没有设置, 我们回退到用于 RPC 的同一配置。

4.1.37. oslo_messaging_rabbit

下表概述了 `/etc/heat/heat.conf` 文件中的 `[oslo_messaging_rabbit]` 组下可用的选项。

表 4.36. oslo_messaging_rabbit

配置选项 = 默认值	类型	Description
amqp_auto_delete = False	布尔值	AMQP 中的自动删除队列。
amqp_durable_queues = False	布尔值	在 AMQP 中使用持久队列。
direct_mandatory_flag = True	布尔值	(DEPRECATED)为直接发送启用/禁用 RabbitMQ 强制标记。直接发送用作回复, 因此在客户端队列不存在时引发 MessageUndeliverable 异常。MessageUndeliverable 异常将用于循环使发送者恢复。此标志已弃用, 且无法停用此功能
enable_cancel_on_failover = False	布尔值	启用 x-cancel-on-ha-failover 标志, 使 rabbitmq 服务器将取消并通知 consumerswhen 队列停机
heartbeat_in_pthread = False	布尔值	默认情况下, 通过原生 python 线程运行健康检查心跳线程。如果此选项等于 False, 则健康检查 heartbeat 将继承父进程的执行模型。例如, 如果父进程使用 eventlet/greenlet 对 stdlib 进行补丁, 则心跳将通过绿色线程运行。这个选项应该只针对 wsgi 服务设置为 True。
heartbeat_rate = 2	整数值	heartbeat_timeout_threshold 期间的我们检查心跳的频率。

配置选项 = 默认值	类型	Description
heartbeat_timeout_thresh old = 60	整数值	如果 heartbeat 的 keep-alive 失败(0 禁用 heartbeat), 则 Rabbit 代理被视为 down 的秒数。
kombu_compression = None	字符串值	EXPERIMENTAL : Possible 值为 : gzip、bz2。如果没有设置压缩, 则不会使用。这个选项可能在以后的版本中不可用。
kombu_failover_strategy = round-robin	字符串值	确定当当前连接到的 RabbitMQ 节点不可用时如何选择下一个 RabbitMQ 节点。仅在配置中提供多个 RabbitMQ 节点时生效。
kombu_missing_consumer_retry_timeout = 60	整数值	在放弃发送其回复前等待缺少的客户端的时间。这个值不应比 rpc_response_timeout 长。
kombu_reconnect_delay = 1.0	浮点值	响应 AMQP 消费者取消通知前等待多久。
rabbit_ha_queues = False	布尔值	尝试在 RabbitMQ 中使用 HA 队列(x-ha-policy: all)。如果更改这个选项, 您必须擦除 RabbitMQ 数据库。在 RabbitMQ 3.0 中, 在声明队列时, x-ha-policy 参数不再控制队列镜像。如果您只想确保所有队列(带有自动生成的名称的队列除外)都在所有节点上进行镜像, 请运行: "rabbitmqctl set_policy HA ^(?!amq\.) gem {"ha-mode": "all"}"
rabbit_interval_max = 30	整数值	RabbitMQ 连接重试的最大间隔。默认值为 30 秒。
rabbit_login_method = AMQPLAIN	字符串值	RabbitMQ 登录方法。
rabbit_qos_prefetch_count = 0	整数值	指定要预抓取的消息数量。设置为零可允许无限消息。
rabbit_retry_backoff = 2	整数值	连接到 RabbitMQ 时重试之间的 backoff 时长。
rabbit_retry_interval = 1	整数值	使用 RabbitMQ 重试连接的频率。
rabbit_transient_queues_ttl = 1800	整数值	正整数代表队列 TTL (x-expires)的持续时间(以秒为单位)。TTL 期间未使用的队列将被自动删除。参数仅影响 reply 和 fanout 队列。
ssl = False	布尔值	通过 SSL 连接。
<code>`ssl_ca_file = `</code>	字符串值	SSL 证书颁发机构文件(仅在启用了 SSL 时评估)。
<code>`ssl_cert_file = `</code>	字符串值	SSL 证书文件(仅在启用了 SSL 时评估)。

配置选项 = 默认值	类型	Description
<code>`ssl_key_file = `</code>	字符串值	SSL 密钥文件（仅在启用了 SSL 时 valid）。
<code>`ssl_version = `</code>	字符串值	要使用的 SSL 版本（仅在启用了 SSL 时评估）。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1_1 和 TLSv1_2 可能在某些发行版中提供。

4.1.38. oslo_middlewares

下表概述了 `/etc/heat/heat.conf` 文件中的 `[oslo_middlewares]` 组下可用的选项。

表 4.37. oslo_middlewares

配置选项 = 默认值	类型	Description
<code>enable_proxy_headers_parsing = False</code>	布尔值	应用程序是否在代理后面。这决定了中间件是否应该解析标头。
<code>max_request_body_size = 114688</code>	整数值	每个请求的最大正文大小，以字节为单位。
<code>secure_proxy_ssl_header = X-Forwarded-Proto</code>	字符串值	用于确定原始请求协议方案的 HTTP 标头，即使它被 SSL 终止代理隐藏也是如此。

4.1.39. oslo_policy

下表概述了 `/etc/heat/heat.conf` 文件中的 `[oslo_policy]` 组下可用的选项。

表 4.38. oslo_policy

配置选项 = 默认值	类型	描述
<code>enforce_new_defaults = False</code>	布尔值	这个选项控制在评估策略时是否使用旧的弃用默认值。如果为 True ，则不会评估旧的已弃用的默认值。这意味着，如果旧默认值允许任何现有令牌，但不允许使用新默认值，它将被禁止。我们鼓励启用此标志以及 <code>enforce_scope</code> 标志，以便您可以同时获得新默认值和 <code>scope_type</code> 的好处。
<code>enforce_scope = False</code>	布尔值	此选项控制在评估策略时是否强制执行范围。如果为 True ，则请求中使用的令牌范围将与被强制执行策略的 <code>scope_types</code> 进行比较。如果范围不匹配，则引发 <code>InvalidScope</code> 异常。如果为 False ，则会记录一条信息，告知操作员使用了不匹配的范围调用策略。
<code>policy_default_rule = default</code>	字符串值	默认规则。未找到请求的规则时强制执行。

配置选项 = 默认值	类型	描述
policy_dirs = ['policy.d']	多值	存储策略配置文件的目录。它们可以相对于 config_dir 选项或绝对路径定义的搜索路径中的任何目录。由 policy_file 定义的文件必须存在，才能搜索这些目录。缺少或空目录将被忽略。
policy_file = policy.json	字符串值	将角色映射到给定服务权限的文件的相对或绝对路径。必须使用与配置文件设置相关的相对路径。
remote_content_type = application/x-www-form-urlencoded	字符串值	用于发送和接收基于 REST 策略检查的数据的内容类型
remote_ssl_ca_cert_file = None	字符串值	基于 REST 的策略检查的 ca cert 文件的绝对路径
remote_ssl_client_cert_file = None	字符串值	基于 REST 的策略检查的客户端证书的绝对路径
remote_ssl_client_key_file = None	字符串值	绝对路径客户端密钥文件 REST 基于策略检查
remote_ssl_verify_server_cert = False	布尔值	基于 REST 的策略检查的服务器身份验证

4.1.40. paste_deploy

下表概述了 /etc/heat/heat.conf 文件中的 **[paste_deploy]** 组下可用的选项。

表 4.39. paste_deploy

配置选项 = 默认值	类型	Description
api_paste_config = api-paste.ini	字符串值	API 粘贴配置文件。
flavor = None	字符串值	要使用的类别。

4.1.41. profiler

下表概述了 /etc/heat/heat.conf 文件中的 **[profiler]** 组下可用的选项。

表 4.40. profiler

配置选项 = 默认值	类型	Description
connection_string = messaging://	字符串值	<p>通知程序后端的连接字符串。</p> <p>默认值为 messaging://，它将通知程序设置为 oslo_messaging。</p> <p>可能的值示例：</p> <ul style="list-style-type: none"> ● messaging:// - 使用 oslo_messaging 驱动程序来发送范围。 ● redis://127.0.0.1:6379 - 使用 redis 驱动程序发送 span。 ● MongoDB://127.0.0.1:27017 - 使用 mongodb 驱动程序进行发送范围。 ● Elasticsearch://127.0.0.1:9200 - 使用 elasticsearch 驱动程序来发送 span。 ● Jaeger://127.0.0.1:6831 - 使用 jaeger tracing 作为发送 span 的驱动程序。
enabled = False	布尔值	<p>为此节点上的所有服务启用性能分析。</p> <p>默认值为 False（完全禁用性能分析功能）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用该功能 ● false：禁用该功能。分析无法通过此项目操作启动。如果性能分析由另一个项目触发，则此项目部分将为空。
es_doc_type = notification	字符串值	elasticsearch 中通知索引的文档类型。
es_scroll_size = 10000	整数值	Elasticsearch 批量分割大型请求。此参数定义每个批处理的最大大小（例如：es_scroll_size=10000）。
es_scroll_time = 2m	字符串值	此参数是一个时间值参数（例如：es_scroll_time=2m），指示参与搜索的节点将维护相关资源以继续和支持它。

配置选项 = 默认值	类型	Description
filter_error_trace = False	布尔值	<p>启用包含 error/exception 的过滤器追踪来分隔的位置。</p> <p>默认值为 False。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用包含 error/exception 的过滤器追踪。 ● false：禁用过滤器。
hmac_keys = SECRET_KEY	字符串值	<p>用于加密上下文数据的 secret 密钥，以用于性能分析。</p> <p>此字符串值应该具有以下格式：<key1>[,<key2>,...<keyn>]，其中每个键都是随机字符串。通过 REST API 触发性能分析的用户必须在 REST API 调用的标头中设置其中一个键，使其包含此节点的性能分析结果。</p> <p>"enabled"标志和"hmac_keys"配置选项都应设置为启用性能分析。此外，为了跨所有服务生成正确的性能分析信息，OpenStack 项目之间至少需要一个关键。这样可确保它可用于客户端生成 trace，其中包含所有可能的资源的信息。</p>
sentinel_service_name = mymaster	字符串值	<p>Redis sentinel 使用服务名称来识别 master redis 服务。此参数定义名称（例如：sentinel_service_name=mymaster）。</p>
socket_timeout = 0.1	浮点值	<p>Redis sentinel 在连接上提供了一个 timeout 选项。此参数定义该超时（例如：socket_timeout=0.1）。</p>
trace_sqlalchemy = False	布尔值	<p>在服务中启用 SQL 请求性能分析。</p> <p>默认值为 False (SQL 请求不会被跟踪)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用 SQL 请求性能分析。每个 SQL 查询都将是 trace 的一部分，可以通过花费的时间来分析。 ● false：禁用 SQL 请求性能分析。花费的时间仅显示在更高级别的操作上。无法以这种方式分析单个 SQL 查询。

4.1.42. revision

下表概述了 /etc/heat/heat.conf 文件中的 [revision] 组下可用的选项。

表 4.41. revision

配置选项 = 默认值	类型	Description
heat_revision = unknown	字符串值	Heat 构建修订.如果您希望单独管理构建修订,您可以将此部分移到其他文件中,并将它添加为另一个配置选项。

4.1.43. ssl

下表概述了 `/etc/heat/heat.conf` 文件中的 `[ssl]` 组下可用的选项。

表 4.42. ssl

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	用于验证连接客户端的 CA 证书文件。
cert_file = None	字符串值	安全启动服务器时要使用的证书文件。
密码 = None	字符串值	设置可用密码列表。值应该是 OpenSSL 密码列表格式的字符串。
key_file = None	字符串值	安全启动服务器时使用的私钥文件。
version = None	字符串值	要使用的 SSL 版本（仅在启用了 SSL 时评估）。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1.1 和 TLSv1.2 可能在某些发行版中提供。

4.1.44. trustee

下表概述了 `/etc/heat/heat.conf` 文件中的 `[trustee]` 组下可用的选项。

表 4.43. trustee

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域,并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域,并在 v2 身份验证中忽略。

配置选项 = 默认值	类型	Description
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
system-scope = None	字符串值	系统操作的范围
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名

4.1.45. 卷

下表概述了 `/etc/heat/heat.conf` 文件中的 `[volumes]` 组下可用的选项。

表 4.44. 卷

配置选项 = 默认值	类型	Description
backups_enabled = True	布尔值	指明是否启用了 <code>cinder-backup</code> 服务。这是一个临时解决方案，直到 <code>cinder-backup</code> 服务变为可发现的问题，请参阅 LP1.10.134856。

第 5 章 IRONIC

下面的章节包含有关 **ironic** 服务中的配置选项的信息。

5.1. IRONIC.CONF

本节包含 `/etc/ironic/ironic.conf` 文件的选项。

5.1.1. DEFAULT

下表概述了 `/etc/ironic/ironic.conf` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
auth_strategy = keystone	字符串值	ironic-api 使用的验证策略不应在生产环境中使用 "noauth", 因为所有身份验证都会被禁用。
backdoor_port = None	字符串值	启用事件程序后端。可接受的值为 0、<port> 和 <start>:<end>, 其中 0 会导致侦听随机的 tcp 端口号; <port> 会导致监听指定的端口号 (如果正在使用该端口, 则不会启用 backdoor); <start>:<end> 会在指定的端口号范围内侦听最小未使用的端口号。所选端口显示在服务的日志文件中。
backdoor_socket = None	字符串值	启用 eventlet 后端, 使用提供的路径作为可接收连接的 unix 套接字。这个选项与 <code>backdoor_port</code> 相互排斥, 该选项只应提供一个。如果提供这两个选项, 则此选项的存在会覆盖该选项的使用。在路径 <code>{pid}</code> 中将被当前进程的 PID 替换。
bindir = \$pybasedir/bin	字符串值	安装 ironic 二进制文件的目录。
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
control_exchange = openstack	字符串值	默认交换, 哪些主题在哪些方面有范围。可以被 <code>transport_url</code> 选项指定的交换名称覆盖。
debug = False	布尔值	如果设置为 true, 则日志级别将设置为 DEBUG, 而不是默认的 INFO 级别。
debug_tracebacks_in_api = False	布尔值	在 API 响应中返回任何错误响应的服务器回溯。警告: 这是不安全的, 不应在生产环境中使用。
default_bios_interface = None	字符串值	用于没有设置 <code>bios_interface</code> 字段的节点的默认 bios 接口。通过枚举 "ironic.hardware.interfaces.bios" 入口点, 可以找到系统上存在的 bios 接口的完整列表。

配置选项 = 默认值	类型	Description
default_boot_interface = None	字符串值	用于没有设置 boot_interface 字段的节点的默认引导接口。通过枚举 "ironic.hardware.interfaces.boot" 入口点，可以找到系统上存在的完整引导接口列表。
default_console_interface = None	字符串值	用于没有设置 console_interface 字段的节点的默认控制台接口。通过枚举 "ironic.hardware.interfaces.console" 入口点，可以找到系统中存在的控制台接口的完整列表。
default_deploy_interface = None	字符串值	用于没有设置 deploy_interface 字段的节点的默认部署接口。通过枚举 "ironic.hardware.interfaces.deploy" 入口点，可以找到系统上存在的部署接口的完整列表。
default_inspect_interface = None	字符串值	用于没有设置 inspect_interface 字段的节点的默认检查接口。通过枚举 "ironic.hardware.interfaces.inspect" 入口点，可以找到系统上存在的检查接口的完整列表。
default_log_levels = ['amqp=WARNING', 'amqp-lib=WARNING', 'qpid.messaging=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'sqlalchemy=WARNING', 'stevedore=INFO', 'eventlet.wsgi.server=INFO', 'iso8601=WARNING', 'requests=WARNING', 'glanceclient=WARNING', 'urllib3.connectionpool=WARNING', 'keystone.middleware.auth_token=INFO', 'keystoneauth.session=INFO', 'openstack=WARNING']	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 log_config_append，则忽略此选项。
default_management_interface = None	字符串值	用于没有设置 management_interface 字段的节点的默认管理界面。通过枚举 "ironic.hardware.interfaces.management" 入口点，可以找到系统中存在的管理界面的完整列表。
default_network_interface = None	字符串值	用于没有设置 network_interface 字段的节点的默认网络接口。通过枚举 "ironic.hardware.interfaces.network" 入口点，可以找到系统上存在的网络接口的完整列表。

配置选项 = 默认值	类型	Description
default_portgroup_mode = active-backup	字符串值	portgroups 的默认模式。允许的值可在 Linux 内核文档绑定中找到： https://www.kernel.org/doc/Documentation/networking/bonding.txt 。
default_power_interface = None	字符串值	用于没有设置 power_interface 字段的节点的默认电源接口。通过枚举 "ironic.hardware.interfaces.power" 入口点，可以找到系统上存在的电源接口的完整列表。
default_raid_interface = None	字符串值	用于没有设置 raid_interface 字段的节点的默认 raid 接口。通过枚举 "ironic.hardware.interfaces.raid" 入口点，可以找到系统上存在的 raid 接口的完整列表。
default_rescue_interface = None	字符串值	用于没有设置 rescue_interface 字段的节点的默认救援接口。通过枚举 "ironic.hardware.interfaces.rescue" 入口点，可以找到系统上存在的救援接口的完整列表。
default_resource_class = None	字符串值	在创建请求中没有提供资源类时用于新节点的资源类。
default_storage_interface = noop	字符串值	用于没有设置 storage_interface 字段的节点的默认存储接口。通过枚举 "ironic.hardware.interfaces.storage" 入口点，可以找到系统中存在的存储接口的完整列表。
default_vendor_interface = None	字符串值	用于没有设置 vendor_interface 字段的节点的默认供应商接口。通过枚举 "ironic.hardware.interfaces.vendor" 入口点，可以找到系统上存在的厂商接口的完整列表。
enabled_bios_interfaces = ['no-bios']	列表值	指定要在服务初始化期间加载的 bios 接口列表。缺少 bios 接口或无法初始化的 bios 接口将阻止 ironic-conductor 服务启动。此处必须启用每个启用的硬件类型支持的 bios 接口，否则 ironic-conductor 服务将不会启动。不能是一个空列表。默认值是一组推荐的面向生产环境的 bios 接口。通过枚举 "ironic.hardware.interfaces.bios" 入口点，可以找到系统上存在的 bios 接口的完整列表。在设置这个值时，请确保每个启用的硬件类型在每个 ironic-conductor 服务中都有相同的启用的 bios 接口。

配置选项 = 默认值	类型	Description
enabled_boot_interfaces = ['pxe']	列表值	指定在服务初始化过程中载入的引导接口列表。缺少引导接口或无法初始化的引导接口将导致 ironic-conductor 服务启动。每个启用的硬件类型支持至少一个引导接口必须在这里启用，或者 ironic-conductor 服务将不会启动。不能是一个空列表。默认值是一组推荐的面向生产环境的引导接口。通过枚举 "ironic.hardware.interfaces.boot" 入口点，可以找到系统上存在的完整引导接口列表。在设置这个值时，请确保每个启用的硬件类型在每个 ironic-conductor 服务中都有相同的启用的引导接口。
enabled_console_interfaces = ['no-console']	列表值	指定要在服务初始化过程中载入的控制台接口列表。缺少控制台接口或无法初始化的控制台接口会阻止 ironic-conductor 服务启动。至少一个由每个启用的硬件类型支持的控制台接口必须在这里启用，否则 ironic-conductor 服务将不会启动。不能是一个空列表。默认值是一组推荐的面向生产环境的控制台接口。通过枚举 "ironic.hardware.interfaces.console" 入口点，可以找到系统中存在的控制台接口的完整列表。在设置这个值时，请确保每个启用的硬件类型在每个 ironic-conductor 服务中都有相同的启用的控制台接口。
enabled_deploy_interfaces = ['direct']	列表值	指定在服务初始化过程中载入的部署接口列表。缺少部署接口或部署无法初始化的接口将导致 ironic-conductor 服务启动。至少一个由每个启用的硬件类型支持的部署接口必须在这里启用，否则 ironic-conductor 服务将不会启动。不能是一个空列表。默认值是一组推荐的面向生产环境的部署接口。通过枚举 "ironic.hardware.interfaces.deploy" 入口点，可以找到系统上存在的部署接口的完整列表。在设置这个值时，请确保每个启用的硬件类型在每个 ironic-conductor 服务中都有相同的启用的部署接口。
enabled_hardware_types = ['ipmi']	列表值	指定在服务初始化过程中载入的硬件类型列表。缺少硬件类型或无法初始化的硬件类型将阻止编排器服务启动。这个选项默认为推荐的一组面向生产环境的硬件类型。通过枚举 "ironic.hardware.types" 入口点，可以找到系统中存在的硬件类型的完整列表。
enabled_inspect_interfaces = ['no-inspect']	列表值	指定在服务初始化过程中载入的检查接口的列表。缺少检查接口，或检查没有初始化的接口，将阻止 ironic-conductor 服务启动。至少一个由每个启用的硬件类型支持的检查接口必须在这里启用，否则 ironic-conductor 服务将不会启动。不能是一个空列表。默认值是一组推荐的面向生产环境的检查接口。通过枚举 "ironic.hardware.interfaces.inspect" 入口点，可以找到系统上存在的检查接口的完整列表。在设置这个值时，请确保每个启用的硬件类型在每个 ironic-conductor 服务中都有相同的启用的检查接口。

配置选项 = 默认值	类型	Description
enabled_management_interfaces = ['ipmitool']	列表值	指定在服务初始化过程中载入的管理接口列表。缺少管理接口或无法初始化的管理接口将阻止 ironic-conductor 服务启动。每个启用的硬件类型支持至少一个管理接口必须在这里启用，否则 ironic-conductor 服务将不会启动。不能是一个空列表。默认值是一组推荐的面向生产环境的管理界面。通过枚举 "ironic.hardware.interfaces.management" 入口点，可以找到系统中存在的管理界面的完整列表。在设置这个值时，请确保每个启用的硬件类型在每个 ironic-conductor 服务中都有相同的启用的管理接口集。
enabled_network_interfaces = ['flat', 'noop']	列表值	指定在服务初始化过程中载入的网络接口列表。缺少网络接口或无法初始化的网络接口将导致 ironic-conductor 服务启动。每个启用的硬件类型支持至少一个网络接口必须在这里启用，否则 ironic-conductor 服务将不会启动。不能是一个空列表。默认值是一组推荐的面向生产环境的网络接口。通过枚举 "ironic.hardware.interfaces.network" 入口点，可以找到系统上存在的网络接口的完整列表。在设置这个值时，请确保每个启用的硬件类型在每个 ironic-conductor 服务中都有相同的启用的网络接口集。
enabled_power_interfaces = ['ipmitool']	列表值	指定在服务初始化过程中载入的电源接口列表。缺少电源接口或无法初始化的电源接口将导致 ironic-conductor 服务启动。每个启用的硬件类型支持至少一个电源接口必须在这里启用，否则 ironic-conductor 服务将不会启动。不能是一个空列表。默认值是一组推荐的面向生产环境的电源接口。通过枚举 "ironic.hardware.interfaces.power" 入口点，可以找到系统上存在的电源接口的完整列表。在设置这个值时，请确保每个启用的硬件类型在每个 ironic-conductor 服务上都有相同的启用电源接口集。
enabled_raid_interfaces = ['agent', 'no-raid']	列表值	指定要在服务初始化过程中载入的 raid 接口列表。缺少 raid 接口，或者无法初始化的 raid 接口将导致 ironic-conductor 服务启动。每个启用的硬件类型支持至少一个 raid 接口必须在这里启用，或者 ironic-conductor 服务将不会启动。不能是一个空列表。默认值是一组推荐的适用于生产环境的 raid 接口。通过枚举 "ironic.hardware.interfaces.raid" 入口点，可以找到系统上存在的 raid 接口的完整列表。在设置这个值时，请确保每个启用的硬件类型在每个 ironic-conductor 服务中都有相同的启用 raid 接口集。

配置选项 = 默认值	类型	Description
enabled_rescue_interfaces = ['no-rescue']	列表值	指定要在服务初始化过程中载入的救援接口列表。缺少救援接口，或者无法初始化的救援接口会阻止 ironic-conductor 服务启动。每个启用的硬件类型支持至少一个救援接口必须在这里启用，或者 ironic-conductor 服务将不会启动。不能是一个空列表。默认值是一组推荐的面向生产环境的救援接口。通过枚举 "ironic.hardware.interfaces.rescue" 入口点，可以找到系统上存在的救援接口的完整列表。在设置这个值时，请确保每个启用的硬件类型在每个 ironic-conductor 服务中都有相同的启用的救援接口。
enabled_storage_interfaces = ['cinder', 'noop']	列表值	指定在服务初始化过程中载入的存储接口列表。缺少存储接口或无法初始化的存储接口将导致 ironic-conductor 服务启动。每个启用的硬件类型支持至少一个存储接口必须在这里启用，或者 ironic-conductor 服务将不会启动。不能是一个空列表。默认值是一组推荐的面向生产环境的存储接口。通过枚举 "ironic.hardware.interfaces.storage" 入口点，可以找到系统中存在的存储接口的完整列表。在设置这个值时，请确保每个启用的硬件类型在每个 ironic-conductor 服务中都有相同的启用的存储接口集。
enabled_vendor_interfaces = ['ipmitool', 'no-vendor']	列表值	指定在服务初始化过程中载入的供应商接口列表。缺少供应商接口或无法初始化的供应商接口将导致 ironic-conductor 服务启动。此处必须启用每个启用的硬件类型支持的供应商接口，否则 ironic-conductor 服务将不会启动。不能是一个空列表。默认值是一组推荐的面向生产环境的供应商接口。通过枚举 "ironic.hardware.interfaces.vendor" 入口点，可以找到系统上存在的厂商接口的完整列表。在设置这个值时，请确保每个启用的硬件类型在每个 ironic-conductor 服务中都有相同的启用的供应商接口。
esp_image = None	字符串值	EFI 系统分区镜像文件的路径。建议有效地创建 UEFI 可引导 ISO 镜像。ESP 镜像应该包含为每个硬件架构 ironic 需要引导的 FAT12/16/32 格式的文件系统（例如 GRUB2）。这个选项只有在没有将 ESP 和 ISO 部署镜像配置为要部署的节点时才使用，在这种情况下，ironic 将尝试从配置的位置获取 ESP 镜像，或者从 UEFI 启动的部署 ISO 镜像中提取 ESP 镜像。
executor_thread_pool_size = 64	整数值	当 executor 为 threading 或 eventlet 时，executor 线程池的大小。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
force_raw_images = True	布尔值	如果为 True，请将后备镜像转换为"原始"磁盘镜像格式。

配置选项 = 默认值	类型	Description
graceful_shutdown_timeout = 60	整数值	指定安全关闭服务器将退出的超时。零值表示无限期等待。
grub_config_path = /boot/grub/grub.cfg	字符串值	由 ironic 生成的 UEFI ISO 镜像中的 GRUB2 配置文件位置。默认值通常是不正确的，不应依赖于。如果您使用特定发行版本中的 GRUB2 镜像，请在此处使用特定于发行版的路径，例如 EFI/ubuntu/grub.cfg
grub_config_template = \$pybasedir/common/grub_conf.template	字符串值	grub 配置文件的模板文件。
hash_partition_exponent = 5	整数值	体验来确定在跨编排器分布负载时要使用的哈希分区数量。在重新平衡环时，较大的值将导致负载的分配和减少负载，但内存使用量越多。每个编排器的分区数为(2^hash_partition_exponent)。这决定了重新平衡的粒度：给定 10 个主机，2 的 exponent，环中有 40 个分区。在大多数情形中，一些分区应该平稳进行重新平衡。默认值适用于最多几百个编排器。为太多分区配置对 CPU 使用率造成负面影响。
hash_ring_algorithm = md5	字符串值	构建哈希环时要使用的哈希函数。如果在 FIPS 系统上运行，请不要使用 md5。警告：集群中的所有 ironic 服务必须始终使用相同的算法。更改算法需要离线更新。
hash_ring_reset_interval = 15	整数值	哈希环被视为过时并在下次访问时刷新的时间（以秒为单位）。
Host = <based on operating system>	字符串值	此节点的名称。这可以是不透明标识符。它不一定是主机名、FQDN 或 IP 地址。但是，节点名称必须在 AMQP 密钥内有效，如果使用 ZeroMQ（将在 Stein 发行版本中删除）、有效的主机名、FQDN 或 IP 地址。
http_basic_auth_user_file = /etc/ironic/htpasswd	字符串值	auth_strategy=http_basic 时使用的 Apache 格式用户身份验证文件的路径
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。
isolinux_bin = /usr/lib/syslinux/isolinux.bin	字符串值	isolinux 二进制文件文件的路径。

配置选项 = 默认值	类型	Description
isolinux_config_template = \$pybasedir/common/isolinux_config.template	字符串值	isolinux 配置文件的模板文件。
ldlinux_c32 = None	字符串值	ldlinux.c32 文件的路径。syslinux 5.0 或更高版本需要这个文件。如果没有指定，则会在 "/usr/lib/syslinux/modules/bios/ldlinux.c32" 和 "/usr/share/syslinux/ldlinux.c32" 中查找该文件。
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 log-date-format）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 %(asctime)s 的格式字符串。默认：%(default)s。如果设置了 log_config_append，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 log_file 路径的基础目录。如果设置了 log_config_append，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 use_stderr 定义的 stderr。如果设置了 log_config_append，则忽略此选项。
log_in_db_max_size = 4096	整数值	任何节点 last_error/maintenance_reason 推送到数据库的最大字符数。
log_options = True	布尔值	启动服务时，启用或禁用所有注册选项的日志值（在 DEBUG 级别）。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 log_rotation_type 设置为 "interval"，否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。

配置选项 = 默认值	类型	Description
logging_context_format_string = <code>%(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s</code>	字符串值	使用上下文格式化用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_debug_format_s uffix = <code>%(funcName)s % (pathname)s:%(lineno)d</code>	字符串值	当消息的日志记录级别为 DEBUG 时，要附加到日志消息的其他数据。 oslo_log.formatters.ContextFormatter 使用
logging_default_format_s tring = <code>%(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] % (instance) s% (message) s</code>	字符串值	上下文未定义时用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_exception_prefix = <code>%(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s</code>	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_for mat = <code>%(user) s %(tenant) s %(domain) s % (user_domain) s % (project_domain) s</code>	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
minimum_memory_wait_r etries = 6	整数值	如果编排器可以自动重试，在失败或将线程返回到池前要暂停的重试次数。
minimum_memory_wait_t ime = 15	整数值	在启动进程前，在重试可用内存之间等待几秒钟。这和 memory_wait_retries 结合使用可让编排器确定我们应尝试直接重试的时长。
minimum_memory_warni ng_only = False	布尔值	将 设置为管理，如果 Ironic 应该只发出警告，而不是尝试返回请求，以防止系统内存耗尽。
minimum_required_mem ory = 1024	整数值	在编排器上启动内存密集型进程前，系统在启动内存密集型进程前，以 MiB 为单元的最小内存。

配置选项 = 默认值	类型	Description
my_ip = <based on operating system>	字符串值	此主机的 IPv4 地址。如果未设置，将以编程方式确定 IP。如果无法这样做，将使用"127.0.0.1"。注意：此字段接受 IPv6 地址作为模板和 URL 的覆盖，但建议使用 [DEFAULT]my_ipv6 以及双栈环境的服务 URL 的 DNS 名称。
my_ipv6 = None	字符串值	此主机的 IP 地址使用 IPv6。这个值必须通过配置提供，不能以编程方式确定 IPv4 的 [DEFAULT]my_ip 参数。
notification_level = None	字符串值	指定要发送通知的最低级别。如果没有设置，则不会发送通知。默认是此选项取消设置。
parallel_image_downloads = False	布尔值	并行运行镜像下载和原始格式转换。
pecan_debug = False	布尔值	启用 pecan 调试模式。警告：这是不安全的，不应在生产环境中使用。
pin_release_version = None	字符串值	用于滚动升级。设置此选项会降级（或固定）Bare Metal API、内部 ironic RPC 通信，以及数据库对象到其相应的版本，以便它们与旧服务兼容。在进行从 N 版本到 N+1 的滚动升级时，将这个设置（固定）为 N。要取消固定（默认），请取消设置，这会使用最新版本。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
pybasedir = /usr/lib/python3.9/site-packages/ironic	字符串值	安装了 ironic python 模块的目录。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
raw_image_growth_factor = 2.0	浮点值	用于估算从精简镜像格式（如 QCOW2）转换的原始镜像大小时的缩放因素。默认值为 2.0，必须大于 1.0。
rootwrap_config = /etc/ironic/rootwrap.conf	字符串值	用于以 root 身份运行命令的 rootwrap 配置文件的路径。

配置选项 = 默认值	类型	Description
rpc_conn_pool_size = 30	整数值	RPC 连接池的大小。
rpc_ping_enabled = False	布尔值	添加端点以回答 ping 调用。端点名为 oslo_rpc_server_ping
rpc_response_timeout = 60	整数值	从调用等待响应的秒数。
rpc_transport = oslo	字符串值	用于在编排器和 API 服务间使用的 RPC 传输实现
run_external_periodic_tasks = True	布尔值	某些定期任务可以在单独的进程中运行。我们是否应该在此处运行？
state_path = \$pybasedir	字符串值	维护 ironic 状态的顶级目录。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
tempdir = /tmp	字符串值	临时工作目录，默认为 Python temp dir。
transport_url = rabbit://	字符串值	<p>用于连接到消息传递后端的网络地址和可选用户凭证，采用 URL 格式。预期格式为：</p> <pre>driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query</pre> <p>示例：rabbit://rabbitmq:password@127.0.0.1:5672//</p> <p>有关 URL 中字段的详情，请查看 <code>oslo_messaging.TransportURL</code> 文档 https://docs.openstack.org/oslo.messaging/latest/reference/transport.html</p>
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。

配置选项 = 默认值	类型	Description
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 <code>log_config_append</code> ，则忽略此选项。
versioned_notifications_topics = ['ironic_versioned_notifications']	列表值	<p>指定 Ironic 发布的版本通知的主题。</p> <p>对于大多数部署，默认值是正常的，很少需要更改。但是，如果您有一个使用版本通知的第三方服务，可能需要获得该服务的一个主题。Ironic 将向此列表中的每个主题队列发送一条包含版本控制通知有效负载的消息。</p> <p>版本通知列表在 https://docs.openstack.org/ironic/latest/admin/notifications.html 中看到</p>
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 <code>log_file</code> 选项并使用 Linux 平台时，才有意义。如果设置了 <code>log_config_append</code> ，则忽略此选项。
webserver_connection_timeout = 60	整数值	使用镜像访问远程 Web 服务器时的连接超时。
webserver_verify_ca = True	字符串值	用于证书验证的 CA 证书。这可以是布尔值或 CA_BUNDLE 文件的路径。如果设为 True，则标准路径中存在的证书用于验证主机证书。如果设为 False，则编排器将忽略验证主机提供的 SSL 证书。如果为 true，则编排器将使用指定的证书进行验证。如果为 SSL 验证，则编排器将使用指定的证书进行验证。如果路径不存在，则行为与这个值设置为 True 时的行为相同。例如，标准路径中存在的证书用于 SSL 验证。Defaults to True.

5.1.2. agent

下表概述了 `/etc/ironic/ironic.conf` 文件中的 **[agent]** 组下可用的选项。

表 5.1. agent

配置选项 = 默认值	类型	Description
agent_api_version = v1	字符串值	用于与 ramdisk 代理通信的 API 版本。
api_ca_file = None	字符串值	用于启动裸机 API 的 TLS CA 的路径。在某些引导方法中，该文件可以传递到 ramdisk。
certificates_path = /var/lib/ironic/certificates	字符串值	存储自动生成的 TLS 证书的路径，用于验证到 ramdisk 的连接。

配置选项 = 默认值	类型	Description
command_timeout = 60	整数值	IPA 命令的超时时间（以秒为单位）。
command_wait_attempts = 100	整数值	在超时前检查异步命令完成的尝试次数。
command_wait_interval = 6	整数值	在检查异步命令完成之间等待的秒数。
deploy_logs_collect = on_failure	字符串值	Ironic 是否应该收集部署失败时的部署日志 (on_failure), 始终或永不。
deploy_logs_local_path = /var/log/ironic/deploy	字符串值	应该存储日志的目录路径, 在 deploy_logs_storage_backend 配置为"local"时使用。
deploy_logs_storage_backend = local	字符串值	存储日志的存储后端的名称。
deploy_logs_swift_container = ironic_deploy_logs_container	字符串值	用于存储日志的 Swift 容器的名称, 在 deploy_logs_storage_backend 配置为"swift"时使用。
deploy_logs_swift_days_to_expire = 30	整数值	日志对象在 Swift 中标记为已过期前的天数。如果 None, 日志将永久保留, 或直到手动删除为止。当 deploy_logs_storage_backend 配置为"swift"时使用。
image_download_source = http	字符串值	指定直接部署接口是否应该尝试直接使用镜像源, 或者 ironic 是否应该缓存编排器中的镜像, 并从 ironic 自己的 http 服务器提供该镜像。
manage_agent_boot = True	布尔值	Ironic 是否将管理代理 ramdisk 的引导。如果设置为 False, 则需要配置您的机制以允许引导代理 ramdisk。
max_command_attempts = 3	整数值	这是为因为网络问题而失败的 IPA 命令执行的最大尝试次数。
memory_consumed_by_agent = 0	整数值	当代理在裸机节点上引导时, 代理消耗的内存大小 (以 MiB 为单位)。这用于检查在引导代理 ramdisk 后是否可以下载并部署到裸机节点上。这可以根据代理 ramdisk 镜像消耗的内存设置。
neutron_agent_max_attempts = 100	整数值	在死代理的网络错误前验证 Neutron 代理状态的最大尝试次数。
neutron_agent_poll_interval = 2	整数值	Neutron 代理在轮询设备更改之间等待的时间。这个值应该与 Neutron 配置中的 CONF.AGENT.polling_interval 相同。

配置选项 = 默认值	类型	Description
neutron_agent_status_retry_interval = 10	整数值	在尝试验证 Neutron 代理状态之间等待时间（以秒为单位）。
post_deploy_get_power_state_retries = 6	整数值	重试获取电源状态的次数，以检查裸机节点是否在软关闭后关闭。
post_deploy_get_power_state_retry_interval = 5	整数值	触发软关闭后轮询电源状态之间等待的时间（以秒为单位）。
require_tls = False	布尔值	如果设置为 True，则没有 https:// 的回调 URL 将由编排器拒绝。
stream_raw_images = True	布尔值	代理 ramdisk 是否应该将原始镜像直接流传输到磁盘上。通过将原始镜像直接流传输到磁盘上，代理 ramdisk 不会将镜像复制到 tmpfs 分区（占用较少的内存）的时间，然后再将其写入磁盘。除非镜像复制为真正慢的磁盘，否则此选项应设置为 True。默认值为 True。
verify_ca = True	字符串值	TLS CA 的路径以验证到 ramdisk 的连接。设置为 True 以使用系统默认 CA 存储。设置为 False 以禁用验证。使用自动 TLS 设置时忽略。

5.1.3. anaconda

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[anaconda]` 组下可用的选项。

表 5.2. anaconda

配置选项 = 默认值	类型	描述
default_ks_template = \$pybasedir/drivers/modules/ks.cfg.template	字符串值	在 instance_info 或 glance OS 镜像中没有指定 kickstart 模板时，要使用的 Kickstart 模板。

5.1.4. ansible

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[ansible]` 组下可用的选项。

表 5.3. ansible

配置选项 = 默认值	类型	Description
ansible_extra_args = None	字符串值	在每次调用 Ansible 时要传递的额外参数。

配置选项 = 默认值	类型	Description
ansible_playbook_script = ansible-playbook	字符串值	到 "ansible-playbook" 脚本的路径。默认将搜索为运行 ironic-conductor 进程的用户配置的 \$PATH。当 ansible-playbook 不在 \$PATH 中或不安装到默认位置时，提供完整路径。
config_file_path = \$pybasedir/drivers/modules/ansible/playbooks/ansible.cfg	字符串值	ansible 配置文件的路径。如果设置为空，则将使用系统默认。
default_clean_playbook = clean.yaml	字符串值	path（相对于 \$playbooks_path 或 absolute）到用于节点清理的默认 playbook。它可能会被节点的 <i>driver_info</i> 字段中的 per-node <i>ansible_clean_playbook</i> 选项覆盖。
default_clean_steps_config = clean_steps.yaml	字符串值	path（相对于 \$playbooks_path 或 absolute）到节点清理过程中使用的默认辅助清理步骤文件。它可能会被节点的 <i>driver_info</i> 字段中的 per-node <i>ansible_clean_steps_config</i> 选项覆盖。
default_deploy_playbook = deploy.yaml	字符串值	path（相对于 \$playbooks_path 或 absolute）到用于部署的默认 playbook。它可能会被节点的 <i>driver_info</i> 字段中的 per-node <i>ansible_deploy_playbook</i> 选项覆盖。
default_key_file = None	字符串值	在通过 SSH 连接到 ramdisk 时，默认情况下 Ansible 使用的专用 SSH 密钥文件的绝对路径。默认为使用为运行 ironic-conductor 服务的用户配置的默认 SSH 密钥。必须使用密码的私钥预加载到 <i>ssh-agent</i> 。它可能会被节点的 <i>driver_info</i> 字段中按节点 <i>ansible_key_file</i> 选项覆盖。
default_python_interpreter = None	字符串值	受管机器上 python 解释器的绝对路径。它可能会被节点的 <i>driver_info</i> 字段中按节点 <i>ansible_python_interpreter</i> 选项覆盖。默认情况下，ansible 使用 /usr/bin/python
default_shutdown_playbook = shutdown.yaml	字符串值	path（相对于 \$playbooks_path 或 absolute）到用于安全关闭节点的默认 playbook 的路径。它可能会被节点的 <i>driver_info</i> 字段中的 per-node <i>ansible_shutdown_playbook</i> 选项覆盖。
default_username = ansible	字符串值	通过 SSH 连接到 ramdisk 时用于 Ansible 的用户名称。它可能会被节点的 <i>driver_info</i> 字段中按节点 <i>ansible_username</i> 选项覆盖。

配置选项 = 默认值	类型	Description
extra_memory = 10	整数值	预期要被节点上的与 Ansible 相关的进程消耗的额外内存量（以 MiB 为单位）。影响镜像是否适合 RAM 的决定。
image_store_cafile = None	字符串值	用于验证到镜像存储的 SSL 连接的特定 CA 捆绑包。如果没有指定，则使用 ramdisk 中的 CA。不被驱动程序包括的默认 playbook 使用。适用于使用自签名证书的环境。
image_store_certfile = None	字符串值	用于 SSL 连接到镜像存储的客户端证书。不被驱动程序包括的默认 playbook 使用。
image_store_insecure = False	布尔值	在下载镜像时跳过验证到镜像存储的 SSL 连接。只有在测试使用自签名证书的环境时，才建议将其设置为 "True"。
image_store_keyfile = None	字符串值	用于 SSL 连接到镜像存储的客户端密钥。不被驱动程序包括的默认 playbook 使用。
playbooks_path = \$pybasedir/drivers/modules/ansible/playbooks	字符串值	带有 playbook、角色和本地清单的目录路径。
post_deploy_get_power_state_retries = 6	整数值	重试获取电源状态的次数，以检查裸机节点是否在软关闭后关闭。值 0 表示不会在失败时重试。
post_deploy_get_power_state_retry_interval = 5	整数值	触发软关闭后轮询电源状态之间等待的时间（以秒为单位）。
verbosity = None	整数值	在调用 "ansible-playbook" 命令时设置请求的 ansible 详细程度。4 包括详细的 SSH 会话日志记录。启用全局调试时默认为 4，否则默认为 0。

5.1.5. api

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[api]` 组下可用的选项。

表 5.4. api

配置选项 = 默认值	类型	Description
api_workers = None	整数值	OpenStack Ironic API 服务的 worker 数量。默认值等于可用的 CPU 数量，但不超过 4。如果无法检测到 CPU 号，则使用一个 worker。

配置选项 = 默认值	类型	Description
enable_ssl_api = False	布尔值	启用集成的独立 API 到通过 HTTPS 而不是 HTTP 服务请求。如果存在从服务执行 HTTPS 卸载的前端服务，这个选项应该是 False；请注意，您要使用 <code>[oslo_middleware]enable_proxy_headers_parsing</code> 选项启用代理标头解析，或配置 <code>[api]public_endpoint</code> 选项，以设置对 SSL 终止的响应。
host_ip = 0.0.0.0	主机地址值	ironic-api 侦听的 IP 地址或主机名。
max_limit = 1000	整数值	从集合资源在单个响应中返回的最大项目数。
network_data_schema = \$pybasedir/api/controllers/v1/network-data-schema.json	字符串值	此部署所使用的网络数据的 schema。
port = 6385	端口值	ironic-api 侦听的 TCP 端口。
public_endpoint = None	字符串值	构建 API 资源链接时使用的公共 URL（如 "https://ironic.rocks:6384"）。如果 None，则使用请求的主机 URL 构建链接。如果 API 在代理后面运行，则需要更改它来代表代理的 URL。默认值为 None。当通过 <code>[oslo_middleware]enable_proxy_headers_parsing</code> 选项启用代理标头解析时忽略。
ramdisk_heartbeat_timeout = 300	整数值	代理心跳的最大间隔（以秒为单位）。
restrict_lookup = True	布尔值	是否将查找 API 限制为处于特定状态的节点。

5.1.6. audit

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[audit]` 组下可用的选项。

表 5.5. audit

配置选项 = 默认值	类型	Description
audit_map_file = /etc/ironic/api_audit_map.conf	字符串值	ironic-api 服务的审计映射文件的路径。仅在启用 API Audit 时使用。
enabled = False	布尔值	启用 API 请求的审计（用于 ironic-api 服务）。

配置选项 = 默认值	类型	Description
<code>ignore_req_list = `</code>	字符串值	在审计日志记录过程中忽略的 Ironic REST API HTTP 方法的逗号分隔列表。例如：如果将其设定为 "GET,POST", 则不会在任何 GET 或 POST 请求上进行审计。只有在启用 API 审计时才使用它。

5.1.7. cinder

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[cinder]` 组下可用的选项。

表 5.6. cinder

配置选项 = 默认值	类型	Description
<code>action_retries = 3</code>	整数值	失败操作时重试次数（当前仅在分离卷时使用）。
<code>action_retry_interval = 5</code>	整数值	如果失败动作，重试间隔（以秒为单位）（只重试特定的操作）。
<code>auth-url = None</code>	字符串值	身份验证 URL
<code>auth_type = None</code>	字符串值	要加载的身份验证类型
<code>cafile = None</code>	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
<code>certFile = None</code>	字符串值	PEM 编码的客户端证书证书文件
<code>collect-timing = False</code>	布尔值	收集每个 API 调用计时信息。
<code>connect-retries = None</code>	整数值	应该尝试连接错误的最大重试次数。
<code>connect-retry-delay = None</code>	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
<code>default-domain-id = None</code>	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
<code>default-domain-name = None</code>	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
<code>domain-id = None</code>	字符串值	范围的域 ID
<code>domain-name = None</code>	字符串值	范围为的域名

配置选项 = 默认值	类型	Description
endpoint-override = None	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version , min-version , 和/或 max-version 选项。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
max-version = None	字符串值	给定 API 的最大主要版本，旨在作为带有 min_version 的范围的上限。与版本相互排斥。
min-version = None	字符串值	给定 API 的最低主要版本，旨在作为 max_version 的范围低限。与版本相互排斥。如果给出了 min_version，且没有 max_version，它就如同 max version 为 "latest"。
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
retries = 3	整数值	在请求失败的情况下，客户端重试。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = volumev3	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
system-scope = None	字符串值	系统操作的范围

配置选项 = 默认值	类型	Description
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。
version = None	字符串值	给定主版本 API 版本内用于端点 URL 发现的最低 Major API 版本。与 <code>min_version</code> 和 <code>max_version</code> 相互排斥

5.1.8. conductor

下表概述了 `/etc/ironic/ironic.conf` 文件中的 **[conductor]** 组下可用的选项。

表 5.7. conductor

配置选项 = 默认值	类型	Description
allow_deleting_available_nodes = True	布尔值	允许删除处于 <i>available</i> 状态的节点。默认值为 <code>True</code> 。
allow_provisioning_in_maintenance = True	布尔值	在处于维护模式时，是否允许节点进入或进行部署或清理。如果此选项设置为 <code>False</code> ，且节点在部署或清理过程中进入维护，则该过程将在下一个心跳后中止。自动清理或使节点可用也会失败。如果为 <code>True</code> （默认），进程将在节点启动心跳后开始并暂停。将其从维护中移动会导致进程继续进行。

配置选项 = 默认值	类型	Description
automated_clean = True	布尔值	启用或禁用自动清理。自动清理是一组可配置的步骤，如清除磁盘驱动器，在节点上执行以确保它处于基准状态并准备好部署到其中。这在实例删除后完成，并在从"manageable"过渡到"available"状态的过程中完成。启用后，执行清理节点的具体步骤取决于该节点由哪个驱动程序管理；有关详细信息，请参阅单个驱动程序的文档。注意：引入清理操作会导致实例删除需要更长的时间。在所有租户都信任的环境中（例如，因为只有一个租户），可以安全地禁用此选项。
bootloader = None	字符串值	包含 EFI 引导装载程序的 EFI 系统分区镜像的 glance ID、http:// 或 file:// URL。从内核和 ramdisk 构建 UEFI 引导 ISO 时，此映像将供 ironic 使用。UEFI 从分区镜像引导是必需的。
check_allocations_interval = 60	整数值	检查孤立分配的间隔，以秒为单位。设置为 0 可禁用检查。
check_provision_state_interval = 60	整数值	检查置备超时之间的间隔，以秒为单位。设置为 0 可禁用检查。
check_rescue_state_interval = 60	整数值	救援超时检查间隔（秒）。
clean_callback_timeout = 1800	整数值	超时（秒），以等待来自 ramdisk 执行清理的回调。如果达到超时，节点将置于"clean failed"调配状态。设置为 0 可禁用超时。
clean_step_priority_override = {}	dict 值	优先级为 in-band 和 out-band clean 步骤（以 interface.step_name:priority 格式提供）运行自动清理步骤，如 deploy.erase_devices_metadata:123。可多次指定选项，以定义多个步骤的优先级。如果设置为 0，则在清理过程中不会运行这个特定步骤。如果未设置 inband clean 步骤，将使用 ramdisk 中设置的优先级。
conductor_group = `	字符串值	要加入的编排器组的名称。最多可有 255 个字符，不区分大小写。此编排器将仅管理节点上设置了匹配的"conductor_group"字段的节点。
configdrive_swift_container = ironic_configdrive_container	字符串值	用于存储配置驱动器数据的 Swift 容器的名称。当 configdrive_use_object_store 为 True 时使用。

配置选项 = 默认值	类型	Description
configdrive_swift_temp_url_duration = None	整数值	configdrive 临时 URL 无效超时（以秒为单位）。如果设置了，则默认为 <code>deploy_callback_timeout</code> ，否则默认为 1800 秒。当 <code>configdrive_use_object_store</code> 为 True 时使用。
deploy_callback_timeout = 1800	整数值	从部署 ramdisk 等待回调的超时（秒）。设置为 0 可禁用超时。
deploy_kernel = None	字符串值	默认部署镜像的 kernel 的 Glance ID、http:// 或 file:// URL。
deploy_ramdisk = None	字符串值	默认部署镜像的 initramfs 的 glance ID、http:// 或 file:// URL。
enable_mdns = False	布尔值	是否启用通过多播 DNS 发布 baremetal API 端点。
force_power_state_during_sync = True	布尔值	在 <code>sync_power_state</code> 过程中，应该将硬件电源状态设置为数据库中记录的状态(True)，或者应该根据硬件状态(False)更新数据库。
heartbeat_interval = 10	整数值	编排器核心为ats之间的秒。
heartbeat_timeout = 60	整数值	编排器最后一次检查后的最大时间（以秒为单位）。超过这个时间时，编排器被视为不活跃。
inspect_wait_timeout = 1800	整数值	等待节点检查的超时（秒）。0 - 无限。
node_locked_retry_attempts = 3	整数值	获取节点锁定的尝试次数。
node_locked_retry_interval = 1	整数值	在节点锁定尝试之间休眠的秒数。
periodic_max_workers = 8	整数值	通过定期任务可以同时启动的最大 worker 线程数量。RPC 线程池大小应小于 RPC 线程池大小。
power_failure_recovery_interval = 300	整数值	由于电源同步失败，在检查之前处于维护模式的节点之间的间隔（以秒为单位）。成功检索其电源状态后，节点会自动从维护模式移出。设置为 0 以禁用这个检查。
power_state_change_timeout = 60	整数值	等待电源操作完成的秒数，即 baremetal 节点处于所需的电源状态。如果超时，电源操作被视为失败。

配置选项 = 默认值	类型	Description
power_state_sync_max_retries = 3	整数值	在 sync_power_state 失败过程中，限制 Ironic 应该尝试使用 DB 中的节点电源状态同步硬件节点电源状态的次数
require_rescue_password_hashed = False	布尔值	选项使编排器不会回退到救援密码的未哈希版本，从而允许使用旧的 ironic-python-agent ramdisks 的救援。
rescue_callback_timeout = 1800	整数值	超时（秒）从救援 ramdisk 中等待回调。如果达到超时，节点将置于 "rescue failed" provision 状态。设置为 0 可禁用超时。
rescue_kernel = None	字符串值	默认救援镜像的 kernel 的 glance ID、http:// 或 file:// URL。
rescue_password_hash_algorithm = sha256	字符串值	用于救援密码的密码哈希算法。
rescue_ramdisk = None	字符串值	默认救援镜像的 initramfs 的 glance ID、http:// 或 file:// URL。
send_sensor_data = False	布尔值	启用通过通知总线发送传感器数据消息
send_sensor_data_for_undeployed_nodes = False	布尔值	传感器数据收集的默认值是仅收集已部署机器的数据，但操作员可能需要知道硬件中是否存在故障。当设置为 true 时，当通过 send_sensor_data 设置启用传感器数据收集时，编排器将从所有节点收集传感器信息。
send_sensor_data_interval = 600	整数值	通过通知总线向 ceilometer 发送传感器数据消息之间的秒数。
send_sensor_data_types = ['ALL']	列表值	需要发送到 Ceilometer 的以逗号分隔的量表类型列表。默认值为 "ALL"，它是一个特殊值，表示发送所有传感器数据。
send_sensor_data_wait_timeout = 300	整数值	在允许再次进行定期调用前，等待发送传感器数据定期任务的时间（以秒为单位）。应小于 send_sensor_data_interval 值。
send_sensor_data_workers = 4	整数值	可同时启动用于发送传感器定期任务数据的最大 worker 数量。
soft_power_off_timeout = 600	整数值	软重启和软关闭操作的超时（以秒为单位）。这个值必须始终为正数。

配置选项 = 默认值	类型	Description
sync_local_state_interval = 180	整数值	当编排器加入或离开集群时，现有编排器可能需要更新任何持久的本地状态，因为节点在集群移动时。这个选项控制每个编排器将检查它应该"接管"的节点的频率（以秒为单位）。将它设置为 0（或负值）以完全禁用检查。
sync_power_state_interval = 60	整数值	将节点电源状态同步到数据库（以秒为单位）之间的间隔。设置为 0 以禁用同步。
sync_power_state_workers = 8	整数值	可以同时启动的最大 worker 线程数，以便从定期任务同步节点电源状态。
workers_pool_size = 100	整数值	worker 绿色线程池的大小。请注意，双线程将由编排器本身保留，用于处理核心情况和定期任务。在此之上， sync_power_state_workers 将最多需要 7 个绿色线程，默认值为 8。

5.1.9. 控制台

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[console]` 组下可用的选项。

表 5.8. 控制台

配置选项 = 默认值	类型	Description
kill_timeout = 1	整数值	在发送 SIGKILL 信号前，等待控制台子进程退出的时间（以秒为单位）。
port_range = None	字符串值	可用于在 ironic conductor 主机上运行的控制台代理服务的端口范围，格式为 <code><start>:<stop></code> 。Shellinabox 和 Socat 控制台都使用这个选项
socat_address = \$my_ip	IP 地址值	在 ironic conductor 主机上运行的 Socat 服务的 IP 地址。仅由 Socat 控制台使用。
subprocess_checking_interval = 1	整数值	检查控制台子进程状态的时间间隔（以秒为单位）。
subprocess_timeout = 10	整数值	等待控制台子进程启动的时间（以秒为单位）。
terminal = shellinaboxd	字符串值	串行控制台终端程序的路径。仅由 Shell In A Box 控制台使用。
terminal_cert_dir = None	字符串值	包含用于串行控制台访问的终端 SSL 证书(PEM)的目录。仅由 Shell In A Box 控制台使用。

配置选项 = 默认值	类型	Description
terminal_pid_dir = None	字符串值	用于保存终端 pid 文件的目录。如果没有指定，则使用临时目录。
terminal_timeout = 600	整数值	超时（以秒为单位）使终端会话在不活跃时关闭。设置为 0 可禁用超时。仅由 Socat 控制台使用。

5.1.10. CORS

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[cors]` 组下可用的选项。

表 5.9. CORS

配置选项 = 默认值	类型	Description
allow_credentials = True	布尔值	指明实际请求可以包含用户凭证
allow_headers = []	列表值	指定在实际请求过程中可以使用哪些标头字段名称。
allow_methods = ['OPTIONS', 'GET', 'HEAD', 'POST', 'PUT', 'DELETE', 'TRACE', 'PATCH']	列表值	指明在实际请求期间可以使用哪些方法。
allowed_origin = None	列表值	指明此资源是否可以与请求"origin"标头中收到的域共享。格式： <code>"<protocol>://<host>[:<port>]"</code> ，没有尾部斜杠。示例： https://horizon.example.com
expose_headers = []	列表值	指明哪些标头可以安全地公开给 API。默认为 HTTP Simple Headers。
max_age = 3600	整数值	CORS preflight 请求的最大缓存期限。

5.1.11. database

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[database]` 组下可用的选项。

表 5.10. database

配置选项 = 默认值	类型	Description
backend = sqlalchemy	字符串值	用于数据库的后端。
connection = None	字符串值	用于连接到数据库的 SQLAlchemy 连接字符串。
connection_debug = 0	整数值	SQL 调试信息的详细程度：0=None, 100=Everything。

配置选项 = 默认值	类型	Description
<code>`connection_parameters = `</code>	字符串值	用于连接时附加到连接 URL 的可选 URL 参数；指定 <code>param1=value1&param2=value2&...</code>
<code>connection_recycle_time = 3600</code>	整数值	连接池中存在的连接的时间超过这个秒数，下一次从池中签出它们的时间将替换为新的连接。
<code>connection_trace = False</code>	布尔值	将 Python 堆栈跟踪作为注释字符串添加到 SQL。
<code>db_inc_retry_interval = True</code>	布尔值	如果为 True，将数据库操作的重试间隔增加到 <code>db_max_retry_interval</code> 。
<code>db_max_retries = 20</code>	整数值	在引发错误前出现连接错误或死锁错误时的最大重试次数。设置为 -1 以指定无限重试计数。
<code>db_max_retry_interval = 10</code>	整数值	如果设置了 <code>db_inc_retry_interval</code> ，则数据库操作重试之间的最大秒数。
<code>db_retry_interval = 1</code>	整数值	数据库事务重试之间的秒数。
<code>max_overflow = 50</code>	整数值	如果设置，请将这个值用于 SQLAlchemy 的 <code>max_overflow</code> 。
<code>max_pool_size = 5</code>	整数值	在池中保留的最大 SQL 连接数。将值设为 0 表示没有限制。
<code>max_retries = 10</code>	整数值	启动期间数据库连接重试的最大数量。设置为 -1 以指定无限重试计数。
<code>mysql_enable_ndb = False</code>	布尔值	如果为 True，则透明地启用对处理 MySQL 集群 (NDB) 的支持。
<code>mysql_engine = InnoDB</code>	字符串值	要使用的 MySQL 引擎。
<code>mysql_sql_mode = TRADITIONAL</code>	字符串值	用于 MySQL 会话的 SQL 模式。这个选项（包括默认选项）会覆盖任何 <code>server-set</code> SQL 模式。要使用服务器配置设置的任何 SQL 模式，请将其设置为 <code>no</code> 值。 示例： <code>mysql_sql_mode=</code>
<code>pool_timeout = None</code>	整数值	如果设置，请将这个值用于 SQLAlchemy 的 <code>pool_timeout</code> 。
<code>retry_interval = 10</code>	整数值	重新尝试打开 SQL 连接之间的间隔。
<code>slave_connection = None</code>	字符串值	用于连接到从数据库的 SQLAlchemy 连接字符串。

配置选项 = 默认值	类型	Description
sqlite_synchronous = True	布尔值	如果为 True，则 SQLite 使用同步模式。
use_db_reconnect = False	布尔值	启用在连接丢失时进行数据库重新连接的实验性使用。

5.1.12. deploy

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[deploy]` 组下可用的选项。

表 5.11. deploy

配置选项 = 默认值	类型	Description
configdrive_use_object_store = False	布尔值	是否将配置驱动器上传到对象存储。将这个选项设置为 True，将配置驱动器存储在 swift 端点中。
continue_if_disk_secure_erase_fails = False	布尔值	在 Ironic Python Agent 中清理过程中，定义在安全擦除操作(NVMe 或 ATA)失败时该怎么办。如果为 False，清理操作将失败，节点将处于 干净的失败状态 。如果为 True，则继续调用并清理 shred。
create_configuration_priority = None	整数值	通过 Ironic Python Agent ramdisk 从设备创建 RAID 配置的优先级。如果未设置，将使用 ramdisk 中设置的优先级（对于 GenericHardwareManager，默认为 0）。如果设置为 0，则在清理过程中不会运行。
default_boot_mode = bios	字符串值	在节点的 <code>driver_info</code> 、功能或 instance_info 配置中没有请求引导模式时，要使用的默认引导模式。目前默认引导模式为"bios"，但将在以后将其更改为"定义"。建议为这个选项设置明确的值。这个选项仅在管理界面支持引导模式管理时才生效
default_boot_option = local	字符串值	在节点的 <code>driver_info</code> 中不请求引导选项时，使用默认引导选项。默认为 "local"。在 Ussuri 发布之前，默认值为 "netboot"。
delete_configuration_priority = None	整数值	通过 Ironic Python Agent ramdisk 从设备中删除 RAID 配置的优先级。如果未设置，将使用 ramdisk 中设置的优先级（对于 GenericHardwareManager，默认为 0）。如果设置为 0，则在清理过程中不会运行。
disk_eraser_concurrency = 1	整数值	定义 Ironic Python Agent ramdisk 擦除磁盘设备的目标池大小。为擦除磁盘创建的线程数量不能超过这个值或要擦除的磁盘数量。
enable_ata_secure_erase = True	布尔值	是否在清理过程中支持使用 ATA 安全 Erase。默认值为 True。

配置选项 = 默认值	类型	Description
enable_nvme_secure_erase = True	布尔值	是否在清理过程中支持使用 NVMe Secure Erase。目前，user-data 和 crypto 模式支持 nvme-cli format 命令，具体取决于设备 capabilities.Defaults 为 True。
erase_devices_metadata_priority = None	整数值	通过 Ironic Python Agent ramdisk 从设备中删除元数据的优先级，运行带清理的步骤。如果未设置，将使用 ramdisk 中设置的优先级（对于 GenericHardwareManager，默认为 99）。如果设置为 0，则在清理过程中不会运行。
erase_devices_priority = None	整数值	通过 Ironic Python Agent ramdisk 运行带外清除设备的优先级。如果未设置，将使用 ramdisk 中设置的优先级（对于 GenericHardwareManager，默认为 10）。如果设置为 0，则在清理过程中不会运行。
erase_skip_read_only = False	布尔值	如果 ironic-python-agent 在运行"erase_devices"清理步骤时跳过只读设备，其中块设备为零。这需要 ironic-python-agent 6.0.0 或更高版本。默认情况下，只读设备将导致基于非元数据的清理操作失败，因为数据在裸机节点部署之间可能的操作安全风险被保留。
external_callback_url = None	字符串值	用于引导方法的裸机 API 的代理回调 URL，如虚拟介质，可以在 provisioning 网络外提供镜像。默认为来自 [service_catalog] 的配置。
external_http_url = None	字符串值	ironic-conductor 节点的 HTTP 服务器的 URL 用于虚拟介质等引导方法，其中镜像可以在 provisioning 网络外提供。使用 Swift 时不适用。默认为 http_url。
fast_track = False	布尔值	是否允许部署代理在机器生命周期的初始状态执行查找，并传递 ramdisk 的常规设置流程。此功能还启用电源操作，它们是要绕过部署过程的一部分，如果 ramdisk 使用了 fast_track_timeout 设置执行心跳操作。
fast_track_timeout = 300	整数值	对于快速跟踪序列，最后一个心跳事件被视为有效的秒数。此设置通常应小于 "Power-On Self Test" 和典型的 ramdisk 启动率的秒数。这个值不应超过 [api]ramdisk_heartbeat_timeout 设置。
http_image_subdir = agent_images	字符串值	ironic-conductor 节点的 HTTP 根路径下的子目录名称，用于为直接部署接口放置实例镜像，当本地 HTTP 服务合并到提供实例镜像而不是 swift tempurls 时。
http_root = /httpboot	字符串值	ironic-conductor 节点的 HTTP 根路径。

配置选项 = 默认值	类型	Description
http_url = None	字符串值	ironic-conductor 节点的 HTTP 服务器 URL。示例： http://192.12.3:8080
power_off_after_deploy_failure = True	布尔值	在部署失败后是否关闭节点。默认值为 True。
ramdisk_image_download_source = local	字符串值	指定引导 iso 镜像是否应该直接使用镜像源 url 从其原始位置提供，或者 ironic 应该缓存编排器中的镜像并从 ironic 自身的 http 服务器提供它。
shred_final_overwrite_with_zeros = True	布尔值	在编写随机数据后，是否将零写入节点的块设备。即使 <code>deploy.shred_random_overwrite_iterations</code> 为 0，这也会向设备写入零。只有在设备无法为 ATA 安全 Erased 时才使用此选项。默认值为 True。
shred_random_overwrite_iterations = 1	整数值	在 shred 期间，使用随机数据覆盖所有块设备 N 次。这只有在无法 ATA 安全 Erased 时使用。默认为 1。

5.1.13. dhcp

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[dhcp]` 组下可用的选项。

表 5.12. dhcp

配置选项 = 默认值	类型	Description
dhcp_provider = neutron	字符串值	要使用的 DHCP 提供程序。"neutron"使用 Neutron，而"none"则使用 no-op 提供程序。

5.1.14. disk_partitioner

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[disk_partitioner]` 组下可用的选项。

表 5.13. disk_partitioner

配置选项 = 默认值	类型	Description
check_device_interval = 1	整数值	在 Ironic 完成分区表后，它会继续在此间隔上检查附加的 iSCSI 设备状态的活动，然后再将镜像复制到节点，以秒为单位
check_device_max_retries = 20	整数值	检查设备没有被另一个进程访问的次数上限。如果设备在那后仍然忙碌，磁盘分区将被视为已失败。

5.1.15. disk_utils

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[disk_utils]` 组下可用的选项。

表 5.14. disk_utils

配置选项 = 默认值	类型	Description
<code>bios_boot_partition_size = 1</code>	整数值	在 BIOS 中为本地引导时，以 MiB 为单位的 BIOS 引导分区大小（以 MiB 为单位）。
<code>dd_block_size = 1M</code>	字符串值	写入节点磁盘的块大小。
<code>efi_system_partition_size = 200</code>	整数值	为本地引导配置 UEFI 系统时，以 MiB 为单位的 EFI 系统分区大小。
<code>image_convert_attempts = 3</code>	整数值	镜像转换的尝试次数。
<code>image_convert_memory_limit = 2048</code>	整数值	MiB 中的 "qemu-img convert" 的内存限值。通过地址空间资源限制实施。
<code>partition_detection_attempts = 3</code>	整数值	最多尝试检测新创建的分区。
<code>partprobe_attempts = 10</code>	整数值	尝试读取分区的最大尝试次数。

5.1.16. drac

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[drac]` 组下可用的选项。

表 5.15. drac

配置选项 = 默认值	类型	描述
<code>bios_factory_reset_timeout = 600</code>	整数值	等待工厂重置 BIOS 设置完成的最长时间（以秒为单位）。
<code>boot_device_job_status_timeout = 30</code>	整数值	等待引导设备配置作业过渡到正确的状态的最长时间（以秒为单位），以允许重新引导或开机完成。
<code>config_job_max_retries = 240</code>	整数值	配置作业成功完成的最大重试次数。
<code>query_import_config_job_status_interval = 60</code>	整数值	检查完成导入配置任务之间等待的秒数
<code>query_raid_config_job_status_interval = 120</code>	整数值	定期 RAID 作业状态检查间隔（以秒为单位），以确定异步 RAID 配置是否已成功完成。

5.1.17. Glance

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[glance]` 组下可用的选项。

表 5.16. Glance

配置选项 = 默认值	类型	Description
<code>allowed_direct_url_schemes = []</code>	列表值	可以直接通过 <code>direct_url</code> 下载的 URL 方案列表。目前支持的方案： <code>[file]</code> 。
<code>auth-url = None</code>	字符串值	身份验证 URL
<code>auth_type = None</code>	字符串值	要加载的身份验证类型
<code>cafile = None</code>	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
<code>certFile = None</code>	字符串值	PEM 编码的客户端证书文件
<code>collect-timing = False</code>	布尔值	收集每个 API 调用计时信息。
<code>connect-retries = None</code>	整数值	应该尝试连接错误的最大重试次数。
<code>connect-retry-delay = None</code>	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
<code>default-domain-id = None</code>	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
<code>default-domain-name = None</code>	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
<code>domain-id = None</code>	字符串值	范围的域 ID
<code>domain-name = None</code>	字符串值	范围的域名
<code>endpoint-override = None</code>	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 <code>version</code> , <code>min-version</code> , 和/或 <code>max-version</code> 选项。
<code>insecure = False</code>	布尔值	验证 HTTPS 连接。
<code>keyfile = None</code>	字符串值	PEM 编码的客户端证书密钥文件
<code>max-version = None</code>	字符串值	给定 API 的最大主要版本，旨在作为带有 <code>min_version</code> 的范围的上限。与版本相互排斥。

配置选项 = 默认值	类型	Description
min-version = None	字符串值	给定 API 的最低主要版本，旨在作为 max_version 的范围低限。与版本相互排斥。如果给出了 min_version，且没有 max_version，它就如同 max_version 为 "latest"。
num_retries = 0	整数值	从 glance 下载镜像时重试次数。
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = image	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
swift_account = None	字符串值	Glance 用于与 Swift 通信的帐户。格式为 "AUTH_uuid"。"uuid" 是 glance-api.conf 中配置的帐户的 UUID。例如："AUTH_a422b2-91f3-2f46-74b7-d7c9e8958f5d30"。如果没有设置，则根据用于访问 Swift 的项目 ID（如 [swift] 部分中所设置）计算默认值。Swift 临时 URL 格式："endpoint_url/api_version/account/container/object_id"
swift_api_version = v1	字符串值	用于为以下目的创建临时 URL 的 Swift API 版本。默认为 "v1"。Swift 临时 URL 格式："endpoint_url/api_version/account/container/object_id"

配置选项 = 默认值	类型	Description
swift_container = glance	字符串值	Swift 容器 Glance 已配置为将其映像存储在默认为 "glance"，这是 glance-api.conf 中的默认值。Swift 临时 URL 格式： <code>"endpoint_url/api_version/account/container/object_id"</code>
swift_endpoint_url = None	字符串值	用于 <code>"endpoint_url/api_version/account/container/object_id"</code> 的 Swift URL 的 "endpoint, hostname, optional port)。不要包含结尾的"/"。例如，使用 <code>"https://swift.example.com"</code> 。如果使用 RADOS 网关，端点也可以包含 <code>/swift</code> 路径；如果不存在，它将会被附加。用于临时 URL，如果未提供，将从服务目录获取。
swift_store_multiple_containers_seed = 0	整数值	这应该与 Glance 配置文件中同名的配置匹配。当设置为 0 时，单租户存储将仅使用一个容器来存储所有镜像。当设置为 1 到 32 之间的整数值时，单租户存储将使用多个容器来存储镜像，这个值将决定创建多少个容器。
swift_temp_url_cache_enabled = False	布尔值	是否缓存生成的 Swift 临时 URL。只有在使用镜像缓存代理时，将它设置为 true 才有用。默认值为 False。
swift_temp_url_duration = 1200	整数值	临时 URL 有效的时间长度（以秒为单位）。默认值为 20 分钟。如果某些部署在尝试从临时 URL 下载时获得 401 响应代码，请尝试此持续时间。这个值必须大于或等于 <code>swift_temp_url_expected_download_start_delay</code> 的值
swift_temp_url_expected_download_start_delay = 0	整数值	这是部署请求时（以秒为单位）的时间（如果生成 Swift 临时 URL）到 IPA ramdisk 启动并且使用 URL 进行镜像下载的时间。这个值用于检查 Swift 临时 URL 持续时间是否足够大，以便开始下载镜像。另外，如果启用了临时 URL 缓存，这将确定在下载启动时缓存的条目是否仍然有效。 <code>swift_temp_url_duration</code> 值必须大于或等于这个选项的值。默认值为 0。
swift_temp_url_key = None	字符串值	提供给 Swift 的 secret 令牌，以允许下载临时 URL。临时 URL 需要。对于 Swift 后端，默认使用服务项目上的密钥（如 <code>[swift]</code> 部分中所设置）。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称

配置选项 = 默认值	类型	Description
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。
version = None	字符串值	给定主版本 API 版本内用于端点 URL 发现的最低 Major API 版本。与 min_version 和 max_version 相互排斥

5.1.18. healthcheck

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[healthcheck]` 组下可用的选项。

表 5.17. healthcheck

配置选项 = 默认值	类型	Description
backends = []	列表值	可执行健康检查并报告该信息作为请求的一部分的其他后端。
detailed = False	布尔值	显示响应中更详细的信息。安全备注：启用此选项可能会公开有关被监控服务的敏感详情。务必验证它是否不会违反您的安全策略。
disable_by_file_path = None	字符串值	检查是否存在文件，以确定应用是否在端口上运行。DisableByFileHealthcheck 插件使用。
disable_by_file_paths = []	列表值	根据端口检查文件是否存在，以确定应用是否在端口上运行。需要字符串的 "port:path" 列表。DisableByFilesPortsHealthcheck 插件使用。
enabled = False	布尔值	在 <code>/healthcheck</code> 中启用健康检查端点。请注意，这是未经身份验证的。有关更多信息，请参见 https://docs.openstack.org/oslo.middleware/latest/reference/healthcheck_plugins.html 。

配置选项 = 默认值	类型	Description
path = /healthcheck	字符串值	响应 healthcheck 请求的路径。

5.1.19. ilo

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[ilo]` 组下可用的选项。

表 5.18. ilo

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	用于验证 iLO 的 CA 证书文件。
clean_priority_clear_secure_boot_keys = 0	整数值	<code>clear_secure_boot_keys</code> 清理步骤的优先级。默认情况下不启用此步骤。它可启用以清除使用 iLO 注册的所有安全引导密钥。
clean_priority_reset_bios_to_default = 10	整数值	<code>reset_bios_to_default</code> 清理步骤的优先级。
clean_priority_reset_ilo = 0	整数值	<code>reset_ilo clean</code> 步骤的优先级。
clean_priority_reset_ilo_credential = 30	整数值	<code>reset_ilo_credential</code> 清理步骤的优先级。此步骤要求在节点的 <code>driver_info</code> 中使用新密码更新 <code>"ilo_change_password"</code> 参数。
clean_priority_reset_secure_boot_keys_to_default = 20	整数值	<code>reset_secure_boot_keys</code> 清理步骤的优先级。此步骤会将安全引导密钥重置为 <code>manufacturing</code> 默认值。
client_port = 443	端口值	用于 iLO 操作的端口
client_timeout = 60	整数值	iLO 操作的超时时间（以秒为单位）
default_boot_mode = auto	字符串值	在节点的 <code>"properties/capabilities"</code> 功能中没有提供 <code>"boot_mode"</code> 功能时，用于置备的默认引导模式。默认为 <code>"auto"</code> 用于向后兼容。当指定 <code>"auto"</code> 时，将根据系统上的引导模式设置选择默认引导模式。
file_permission = 420	整数值	<code>swift-less</code> 镜像的文件权限，使用文件访问权限的八进制权限表示。此设置默认为 <code>644</code> ，或 Python 中的八进制数 <code>0o644</code> 。此设置必须设置为八进制数表示，即以 <code>0o</code> 开始。

配置选项 = 默认值	类型	Description
kernel_append_params = nofb nomodeset vga=normal	字符串值	传递给实例内核的其他内核参数。这些参数可由内核或应用程序使用，方法是读取 <code>/proc/cmdline</code> 。严重的 <code>cmdline</code> 大小限制！可以被 instance_info/kernel_append_params 属性覆盖。
oob_erase_devices_job_status_interval = 300	整数值	定期擦除状态检查之间的间隔（以秒为单位），以确定异步外除擦除设备是否已成功完成。平均来说，具有默认模式"overwrite"的 300GB HDD 大约需要 9 小时，默认模式为 "block" 的 300GB SSD 需要约为 "block"。30 秒才能完成 sanitize 磁盘清除。
power_wait = 2	整数值	在电源操作之间等待的时间（以秒为单位）
swift_ilo_container = ironic_ilo_container	字符串值	用于存储数据的 Swift iLO 容器。
swift_object_expiry_timeout = 900	整数值	Swift 对象自动过期的时间（以秒为单位）。
use_web_server_for_images = False	布尔值	把它设置为 True，以使用 http web 服务器托管软盘镜像和生成的启动 ISO。这要求 <code>http_root</code> 和 <code>http_url</code> 在配置文件的 <code>[deploy]</code> 部分中配置。如果这设置为 False，则 Ironic 将使用 Swift 托管软盘镜像和生成的 <code>boot_iso</code> 。
verify_ca = True	字符串值	用于验证 iLO 的 CA 证书。这可以是布尔值，也可以是 <code>CA_BUNDLE</code> 文件的路径，也可以是带有可信 CA 证书的目录。如果设置为 True，则驱动程序将验证主机证书；如果 False，则驱动程序将忽略验证 SSL 证书。如果是路径，则驱动程序将使用指定的证书或目录中的其中一个证书。默认值为 True。

5.1.20. inspector

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[inspector]` 组下可用的选项。

表 5.19. inspector

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。

配置选项 = 默认值	类型	Description
callback_endpoint_override = None	字符串值	引导由 ironic 管理时用于回发内省数据的回调端点。默认使用标准 keystoneauth 选项。
certFile = None	字符串值	PEM 编码的客户端证书证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
connect-retries = None	整数值	应该尝试连接错误的最大重试次数。
connect-retry-delay = None	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
endpoint-override = None	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version , min-version , 和/或 max-version 选项。
extra_kernel_params = `	字符串值	引导由 ironic（而非 ironic-inspector）管理时，要传递给检查 ramdisk 的额外内核参数。对键=值用空格分开。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
max-version = None	字符串值	给定 API 的最大主要版本，旨在作为带有 min_version 的范围的上限。与版本相互排斥。
min-version = None	字符串值	给定 API 的最低主要版本，旨在作为 max_version 的范围低限。与版本相互排斥。如果给出了 min_version，且没有 max_version，它就如同 max_version 为 "latest"。
password = None	字符串值	用户的密码

配置选项 = 默认值	类型	Description
power_off = True	布尔值	检查完成后是否关闭节点
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
require_managed_boot = False	布尔值	要求带外检查引导由 ironic 完全管理。如果您安装的 ironic-inspector 没有单独的 PXE 引导环境，则将其设置为 True。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = baremetal-introspection	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
status_check_period = 60	整数值	检查检查时节点状态的时间（以秒为单位）
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID

配置选项 = 默认值	类型	Description
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。
version = None	字符串值	给定主版本 API 版本内用于端点 URL 发现的最低 Major API 版本。与 <code>min_version</code> 和 <code>max_version</code> 相互排斥

5.1.21. ipmi

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[ipmi]` 组下可用的选项。

表 5.20. ipmi

配置选项 = 默认值	类型	Description
additional_retryable_ipmi_errors = []	多值	ipmitool 可能会遇到的额外错误，特定于它在其中运行的环境。
cipher_suite_versions = []	列表值	如果没有为节点设置字段 <code>cipher_suite</code> ，则硬件可以支持的可能密码套件版本列表。
command_retry_timeout = 60	整数值	重试 IPMI 操作的最大时间（以秒为单位）。（例如，如果请求的操作失败，则一个操作可以被重试。）当存在缓慢或无响应的 BMC 时，设置此值可能会导致同步电源状态定期任务挂起。
debug = False	布尔值	启用使用额外的调试输出执行所有 ipmi 命令。这是单独的选项，因为在这个模式中，ipmitool 可以记录大量误导文本。
disable_boot_timeout = True	布尔值	Ironic 发送原始 IPMI 命令的默认超时行为，以禁用 60 秒启动超时。将此选项设置为 False 不会发送该命令，默认值为 True。它可以被节点的 <code>driver_info</code> 字段中的 per-node <code>ipmi_disable_boot_timeout</code> 选项覆盖。
kill_on_timeout = True	布尔值	如果 ipmitool 进程在 <code>command_retry_timeout</code> 超时到期后没有退出，则终止由 ironic 调用的 ipmitool 进程来读取节点电源状态。推荐的设置为 True

配置选项 = 默认值	类型	Description
min_command_interval = 5	整数值	发送到服务器的 IPMI 操作之间的最短时间（以秒为单位）。有些硬件设置得太低时，可能会导致 BMC 崩溃。推荐的设置为 5 秒。
use_ipmitool_retries = False	布尔值	当设置为 True 且 ipmitool 支持参数时，重试间隔的数量和重试间隔会作为参数传递给 ipmitool，ipmitool 将执行重试。当设置为 False 时，ironic 将重试 ipmitool 命令。推荐的设置是 False

5.1.22. irmc

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[irmc]` 组下可用的选项。

表 5.21. irmc

配置选项 = 默认值	类型	Description
auth_method = basic	字符串值	用于 iRMC 操作的验证方法
clean_priority_restore_irmc_bios_config = 0	整数值	restore_irmc_bios_config clean 步骤的优先级。
client_timeout = 60	整数值	iRMC 操作的超时时间（以秒为单位）
fpga_ids = []	列表值	要检查的 CPU FPGA 的厂商 ID 和设备 ID 列表。列表项的格式是 vendorID/deviceID，用逗号分开。CPU 检查将使用此值来查找节点中是否存在 CPU FPGA。如果没有定义这个选项，请在节点特征中保留 CUSTOM_CPU_FPGA。示例 fpga_ids 值： 0x1000/0x0079,0x2100/0x0080
gpu_ids = []	列表值	要检查的 GPU 设备的厂商 ID 和设备 ID 列表。列表项的格式是 vendorID/deviceID，用逗号分开。GPU 检查将使用此值来计算节点中的 GPU 设备数量。如果没有定义这个选项，请在 capabilities 属性中保留 pci_gpu_devices。示例 gpu_ids 值： 0x1000/0x0079,0x2100/0x0080
port = 443	端口值	用于 iRMC 操作的端口
query_raid_config_fgi_status_interval = 300	整数值	定期 RAID 状态检查之间的间隔（以秒为单位），以确定异步 RAID 配置是否已成功完成。在创建虚拟驱动器后，前台初始化(FGI)将启动 5 分钟。
remote_image_server = None	字符串值	远程镜像服务器的 IP

配置选项 = 默认值	类型	Description
remote_image_share_name = share	字符串值	share name of remote_image_server
remote_image_share_root = /remote_image_share_root	字符串值	Ironic 编排器节点的"NFS"或"CIFS"根路径
remote_image_share_type = CIFS	字符串值	共享虚拟介质类型
remote_image_user_domain = `	字符串值	remote_image_user_name 的域名
remote_image_user_name = None	字符串值	remote_image_server 的用户名
remote_image_user_password = None	字符串值	remote_image_user_name 的密码
sensor_method = ipmitool	字符串值	获得感应器数据的方法。
snmp_auth_proto = sha	字符串值	SNMPv3 消息身份验证协议 ID。版本 v3 需要。如果 python-scciclient 的版本为 0.10.1, 则将忽略。有效选项包括 <i>sha</i> , <i>sha256</i> , <i>sha384</i> 和 <i>sha512</i> , 而 <i>sha</i> 是 iRMC S4 和 S5 中唯一支持的协议, 但支持 <i>sha256</i> 、 <i>sha384</i> 和 <i>sha512</i> , 但不支持 <i>sha</i> 。
snmp_community = public	字符串值	SNMP 社区。版本 "v1" 和 "v2c" 需要
snmp_polling_interval = 10	整数值	SNMP 轮询间隔 (以秒为单位)
snmp_port = 161	端口值	SNMP 端口
snmp_priv_proto = aes	字符串值	SNMPv3 消息隐私 (加密) 协议 ID。版本 v3 需要。如果 python-scciclient 的版本为 0.10.1, 则将忽略。支持 AES。
snmp_security = None	字符串值	SNMP 安全性名称。版本 v3 需要。如果设置了 driver_info/irmc_snmp_user, 则将忽略。
snmp_version = v2c	字符串值	SNMP 协议版本

5.1.23. ironic_lib

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[ironic_lib]` 组下可用的选项。

表 5.22. ironic_lib

配置选项 = 默认值	类型	Description
<code>fatal_exception_format_errors = False</code>	布尔值	在生成异常消息时，使用格式错误（编程错误）。如果为 True，则引发异常；如果 False，请使用未格式化的消息。
<code>root_helper = sudo ironic-rootwrap</code> <code>/etc/ironic/rootwrap.conf</code>	字符串值	前缀为以 root 身份运行的命令的命令。如果没有指定，则不会以 root 用户身份运行任何命令。

5.1.24. iscsi

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[iscsi]` 组下可用的选项。

表 5.23. iscsi

配置选项 = 默认值	类型	Description
<code>conv_flags = None</code>	字符串值	需要发送到 dd 命令的标记，以控制复制到主机时原始文件的转换。它可以包含多个选项，用逗号分开。
<code>portal_port = 3260</code>	端口值	iSCSI 门户侦听传入连接的端口号。
<code>verify_attempts = 3</code>	整数值	最多尝试验证 iSCSI 连接是否处于活动状态，在尝试之间休眠 1 秒。默认值为 3。

5.1.25. json_rpc

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[json_rpc]` 组下可用的选项。

表 5.24. json_rpc

配置选项 = 默认值	类型	Description
<code>auth-url = None</code>	字符串值	身份验证 URL
<code>auth_strategy = None</code>	字符串值	JSON RPC 使用的验证策略。默认为全局 <code>auth_strategy</code> 设置。
<code>auth_type = None</code>	字符串值	要加载的身份验证类型
<code>cafile = None</code>	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
<code>certFile = None</code>	字符串值	PEM 编码的客户端证书文件

配置选项 = 默认值	类型	Description
collect-timing = False	布尔值	收集每个 API 调用计时信息。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
host_ip = ::	主机地址值	JSON RPC 将侦听的 IP 地址或主机名。
http_basic_auth_user_file = /etc/ironic/htpasswd-json-rpc	字符串值	auth_strategy=http_basic 时使用的 Apache 格式用户身份验证文件的路径
http_basic_password = None	字符串值	用于 HTTP 基本身份验证客户端请求的密码。
http_basic_username = None	字符串值	用于 HTTP 基本身份验证客户端请求的用户名。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
password = None	字符串值	用户的密码
port = 8089	端口值	JSON RPC 使用的端口
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
split-loggers = False	布尔值	将请求记录到多个日志记录器。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID

配置选项 = 默认值	类型	Description
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
use_ssl = False	布尔值	是否为 JSON RPC 使用 TLS
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名

5.1.26. keystone_authtoken

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[keystone_authtoken]` 组下可用的选项。

表 5.25. keystone_authtoken

配置选项 = 默认值	类型	Description
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
auth_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。这个选项已弃用，现在使用 <code>www_authenticate_uri</code> ，并将在 S 发行版本中删除。弃用自：Queens *reason:* <code>auth_uri</code> 选项已弃用，现在使用 <code>www_authenticate_uri</code> ，并将在 S 发行版本中删除。
auth_version = None	字符串值	Identity API 端点的 API 版本。

配置选项 = 默认值	类型	Description
cache = None	字符串值	存储 Swift 缓存对象的请求环境密钥。当使用 Swift 缓存部署 auth_token 中间件时，请使用此选项使中间件共享具有 swift 的缓存后端。否则，使用 memcached_servers 选项。
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。默认为系统 CA。
certFile = None	字符串值	如果身份服务器需要客户端证书，则需要此项
delay_auth_decision = False	布尔值	不要在中间件内处理授权请求，而是将授权决策委派给下游 WSGI 组件。
enforce_token_bind = permissive	字符串值	用于控制令牌绑定的使用和类型。可以设置为 "disabled" 以不检查令牌绑定。如果绑定类型是已知服务器的绑定类型，则验证绑定信息，并忽略它（如 "permissive"），但如果绑定类型未知，则令牌将被拒绝。"必需"令牌绑定需要允许任何形式的令牌绑定。最后，令牌中必须存在的绑定方法名称。
http_connect_timeout = None	整数值	请求超时值与身份 API 服务器通信。
http_request_max_retries = 3	整数值	与身份 API 服务器通信时，我们尝试重新连接的次数。
include_service_catalog = True	布尔值	（可选）指示是否设置 X-Service-Catalog 标头。如果为 False，则中间件不会要求在令牌验证时请求服务目录，且不会设置 X-Service-Catalog 标头。
insecure = False	布尔值	验证 HTTPS 连接。
interface = internal	字符串值	用于 Identity API 端点的接口。有效值为 "public", "internal"（默认）或 "admin"。
keyfile = None	字符串值	如果身份服务器需要客户端证书，则需要此项
memcache_pool_conn_get_timeout = 10	整数值	（可选）操作等待从池中获取 memcached 客户端连接的秒数。
memcache_pool_dead_retry = 300	整数值	（可选）在再次尝试前 memcached 服务器被视为 dead 的秒数。
memcache_pool_maxsize = 10	整数值	（可选）到每个 memcached 服务器的最大打开连接数。

配置选项 = 默认值	类型	Description
memcache_pool_socket_timeout = 3	整数值	(可选) 与 memcached 服务器通信的套接字超时 (以秒为单位)。
memcache_pool_unused_timeout = 60	整数值	(可选) 与 memcached 的连接在池关闭前未使用的秒数。
memcache_secret_key = None	字符串值	(可选, 如果定义了 memcache_security_strategy) 此字符串用于密钥派生。
memcache_security_strategy = None	字符串值	(可选) 如果定义, 则指令牌数据是否应进行身份验证或加密。如果 MAC, 令牌数据在缓存中进行身份验证 (使用 HMAC)。如果 ENCRYPT, 令牌数据会在缓存中加密并进行身份验证。如果该值不是这些选项之一或为空, 则 auth_token 会在初始化时引发异常。
memcache_use_advanced_pool = False	布尔值	(可选) 使用 advanced (事件安全) memcached 客户端池。高级池只能在 python 2.x 下工作。
memcached_servers = None	列表值	(可选) 指定要用于缓存的 memcached 服务器列表。如果未定义, 令牌将会被缓存在进程中。
region_name = None	字符串值	找到身份服务器的区域。
service_token_roles = ['service']	列表值	需要出现在服务令牌中的角色。服务令牌可以请求可以使用过期的令牌, 因此此检查应严格控制只有实际服务应发送此令牌。此处的角色作为 ANY 检查应用, 因此此列表中的任何角色都必须存在。为了向后兼容的原因, 这目前只会影响 allow_expired 检查。
service_token_roles_required = False	布尔值	为了向后兼容的原因, 我们必须允许有效的服务令牌传递, 但不允许将 service_token_roles 检查视为有效。在以后的发行版本中设置此 true 将成为默认设置, 并应启用 (如果可能)。
service_type = None	字符串值	服务的名称或类型 (在服务目录中出现)。这用于验证具有受限访问规则的令牌。
token_cache_time = 300	整数值	为了防止过度努力验证令牌, 中间件会缓存之前可预见的令牌 (以秒为单位)。设置为 -1 可完全禁用缓存。

配置选项 = 默认值	类型	Description
www_authenticate_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点以进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。

5.1.27. mdns

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[mdns]` 组下可用的选项。

表 5.26. mdns

配置选项 = 默认值	类型	Description
interfaces = None	列表值	用于 mDNS 的接口的 IP 地址列表。默认为系统中的所有接口。
lookup_attempts = 3	整数值	查找服务的尝试次数。
params = {}	dict 值	为注册的服务传递的其他参数。
registration_attempts = 5	整数值	注册服务的尝试次数。由于 zeroconf 库中的竞争条件，目前必须大于 1。

5.1.28. metrics

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[metrics]` 组下可用的选项。

表 5.27. metrics

配置选项 = 默认值	类型	Description
agent_backend = noop	字符串值	用于指标代理 ramdisk 的后端。默认可能的后端为 "noop" 和 "statsd"。
agent_global_prefix = None	字符串值	为代理 ramdisk 发送的所有指标名称添加前缀。指标名称的格式是 <code>[global_prefix.][uuid.]</code> <code>[host_name.]prefix.metric_name</code> 。
agent_prepend_host = False	布尔值	将主机名添加到代理 ramdisk 发送的所有指标名称。指标名称的格式是 <code>[global_prefix.][uuid.]</code> <code>[host_name.]prefix.metric_name</code> 。

配置选项 = 默认值	类型	Description
agent_prepend_host_reverse = True	布尔值	将前置的主机值分割为 ".", 将其反向用于代理 ramdisk 发送的指标 (以便更好地匹配域名的反向分级形式)。
agent_prepend_uuid = False	布尔值	将节点的 Ironic uuid 添加到代理 ramdisk 发送的所有指标名称。指标名称的格式是 [global_prefix.][uuid.][host_name.]prefix.metric_name。
backend = noop	字符串值	用于 metrics 系统的后端。
global_prefix = None	字符串值	使用这个值为所有指标名称添加前缀。默认情况下, 没有全局前缀。指标名称的格式是 [global_prefix.][host_name.]prefix.metric_name。
prepend_host = False	布尔值	将主机名添加到所有指标名称。指标名称的格式是 [global_prefix.][host_name.]prefix.metric_name。
prepend_host_reverse = True	布尔值	将前置的主机值分割为 "." 并反向它 (以更好地匹配域名的反向层次结构形式)。

5.1.29. metrics_statsd

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[metrics_statsd]` 组下可用的选项。

表 5.28. metrics_statsd

配置选项 = 默认值	类型	Description
agent_statsd_host = localhost	字符串值	与 statsd 后端一起使用的代理 ramdisk 的主机。这必须可从代理引导的网络访问。
agent_statsd_port = 8125	端口值	与 statsd 后端一起使用的代理 ramdisk 端口。
statsd_host = localhost	字符串值	用于 statsd 后端的主机。
statsd_port = 8125	端口值	与 statsd 后端一起使用的端口。

5.1.30. molds

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[molds]` 组下可用的选项。

表 5.29. molds

配置选项 = 默认值	类型	描述
password = None	字符串值	"http" Basic auth 的密码。默认情况下，设置为空。
retry_attempts = 3	整数值	重试尝试保存或获取配置 molds。
retry_interval = 3	整数值	保存或获取配置折叠的重试间隔。
storage = swift	字符串值	配置 mold 存储位置。支持 "swift" 和 "http"。默认 "swift"。
user = None	字符串值	user for "http" Basic auth。默认情况下，设置为空。

5.1.31. neutron

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[neutron]` 组下可用的选项。

表 5.30. neutron

配置选项 = 默认值	类型	Description
add_all_ports = False	布尔值	选项，可在创建用于置备、清理或救援的端口时将所有端口传输到 neutron。这是在没有分配给端口的 IP 地址的情况下完成的，在某些绑定的网络配置中可能很有用。
auth-url = None	字符串值	身份验证 URL
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书证书文件
cleaning_network = None	字符串值	引导至中用于清理节点的 ramdisk 的 Neutron 网络 UUID 或名称。"neutron" 网络接口需要。如果使用 "flat" 网络接口或 "neutron" DHCP 供应商时，还需要清理节点。如果提供了名称，则必须在所有网络或清理中唯一，否则清理将失败。
cleaning_network_security_groups = []	列表值	在清理节点的过程中要应用的 Neutron 安全组 UUID 列表。可选的 "neutron" 网络接口，不用于 "flat" 或 "noop" 网络接口。如果未指定，则使用默认安全组。
collect-timing = False	布尔值	收集每个 API 调用计时信息。
connect-retries = None	整数值	应该尝试连接错误的最大重试次数。

配置选项 = 默认值	类型	Description
connect-retry-delay = None	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
dhcpv6_stateful_address_count = 4	整数值	为在 DHCPv6-stateful 网络上创建的用于置备、清理、救援或检查的端口分配的 IPv6 地址数量。链加载过程的不同阶段将请求具有不同 CLID/IAID 的地址。由于主机必须保留非标识性标识符多个地址，以确保引导过程的每个步骤都可以成功租期地址。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
endpoint-override = None	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version , min-version , 和/或 max-version 选项。
insecure = False	布尔值	验证 HTTPS 连接。
inspection_network = None	字符串值	要引导到的 ramdisk 的 Neutron 网络 UUID 或名称，以进行节点进行带处理检查。如果提供了名称，则必须在所有网络或检查中唯一，否则检查将失败。
inspection_network_security_groups = []	列表值	在节点检查过程中应用的 Neutron 安全组 UUID 列表。可选的 "neutron" 网络接口，不用 "flat" 或 "noop" 网络接口。如果未指定，则使用默认安全组。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
max-version = None	字符串值	给定 API 的最大主要版本，旨在作为带有 min_version 的范围的上限。与版本相互排斥。
min-version = None	字符串值	给定 API 的最低主要版本，旨在作为 max_version 的范围低限。与版本相互排斥。如果给出了 min_version，且没有 max_version，它就如同 max version 为 "latest"。
password = None	字符串值	用户的密码

配置选项 = 默认值	类型	Description
port_setup_delay = 0	整数值	延迟值，以等待 Neutron 代理为端口设置足够的 DHCP 配置。
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
provisioning_network = None	字符串值	为调配节点引导到的 ramdisk 的 Neutron 网络 UUID 或名称。"neutron"网络接口需要。如果提供了名称，则必须在所有网络或部署中唯一，否则部署将失败。
provisioning_network_security_groups = []	列表值	在置备节点的过程中应用的 Neutron 安全组 UUID 列表。可选的 "neutron" 网络接口，不用于 "flat"或"noop"网络接口。如果未指定，则使用默认安全组。
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
request_timeout = 45	整数值	与 Neutron 交互时请求处理的超时。如果观察到 neutron 端口操作超时，则应增加这个值，因为 neutron 在返回到 API 客户端之前执行预提交验证的时间可能比正常的客户端/服务器交互要长。
rescuing_network = None	字符串值	为救援模式引导 ramdisk 的 Neutron 网络 UUID 或名称。这不是救援 ramdisk 将使用 post-boot iwl-categories 的租户网络用于该网络的网络。如果使用救援模式，则需要 "neutron" 网络接口。它不适用于 "flat" 或 "noop" 网络接口。如果提供了名称，则它在所有网络之间必须是唯一的，否则 rescue 将失败。
rescuing_network_security_groups = []	列表值	在节点救援过程中应用的 Neutron 安全组 UUID 列表。可选的 "neutron" 网络接口，不用于 "flat"或"noop"网络接口。如果未指定，则使用默认安全组。
retries = 3	整数值	DEPRECATED : 当请求失败时，客户端重试。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = network	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。

配置选项 = 默认值	类型	Description
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。
version = None	字符串值	给定主版本 API 版本内用于端点 URL 发现的最低 Major API 版本。与 min_version 和 max_version 相互排斥

5.1.32. nova

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[nova]` 组下可用的选项。

表 5.31. nova

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。

配置选项 = 默认值	类型	Description
certFile = None	字符串值	PEM 编码的客户端证书证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
connect-retries = None	整数值	应该尝试连接错误的最大重试次数。
connect-retry-delay = None	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
endpoint-override = None	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version , min-version , 和/或 max-version 选项。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
max-version = None	字符串值	给定 API 的最大主要版本，旨在作为带有 min_version 的范围的上限。与版本相互排斥。
min-version = None	字符串值	给定 API 的最低主要版本，旨在作为 max_version 的范围低限。与版本相互排斥。如果给出了 min_version，且没有 max_version，它就如同 max version 为 "latest"。
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围

配置选项 = 默认值	类型	Description
project-name = None	字符串值	范围的项目名称
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
send_power_notifications = True	布尔值	当设置为 True 时，它将启用电源状态将回调更改为 nova 的支持。在没有 openstack 计算服务的部署中，此选项应设置为 False。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = compute	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。
version = None	字符串值	给定主版本 API 版本内用于端点 URL 发现的最低 Major API 版本。与 min_version 和 max_version 相互排斥

5.1.33. oslo_concurrency

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[oslo_concurrency]` 组下可用的选项。

表 5.32. oslo_concurrency

配置选项 = 默认值	类型	Description
disable_process_locking = False	布尔值	启用或禁用进程间锁定。
lock_path = None	字符串值	用于锁定文件的目录。为安全起见，指定目录应只由运行需要锁定的进程的用户写入。默认为环境变量 <code>OSLO_LOCK_PATH</code> 。如果使用外部锁定，则必须设置锁定路径。

5.1.34. oslo_messaging_amqp

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[oslo_messaging_amqp]` 组下可用的选项。

表 5.33. oslo_messaging_amqp

配置选项 = 默认值	类型	Description
addressing_mode = dynamic	字符串值	表示驱动程序使用的寻址模式。允许的值： <i>legacy</i> - 使用旧的不可路由寻址； <i>routable</i> - 使用可路由地址； <i>dynamic</i> - 如果消息总线不支持路由，使用 <i>legacy</i> ，如果支持，使用 <i>routable</i> 。
anycast_address = anycast	字符串值	在发送到一组消费者时附加到地址前缀。消息总线用来识别应在消费者之间以轮循方式发送的消息。
broadcast_prefix = broadcast	字符串值	广播到所有服务器时使用的地址前缀
connection_retry_backoff = 2	整数值	每次失败故障转移尝试后，将 <code>connection_retry_interval</code> 增加到这秒数。
connection_retry_interval = 1	整数值	尝试重新连接前暂停的秒数。
connection_retry_interval_max = 30	整数值	<code>connection_retry_interval</code> + <code>connection_retry_backoff</code> 的最大限制
container_name = None	字符串值	AMQP 容器的名称必须全局唯一。默认为生成的 UUID
default_notification_exchange = None	字符串值	通知地址中使用的交换名称。Exchange name resolution priority: <code>Target.exchange</code> if set <code>default_notification_exchange</code> if set else <code>control_exchange</code> if set else <i>notify</i>

配置选项 = 默认值	类型	Description
default_notify_timeout = 30	整数值	发送通知消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_reply_retry = 0	整数值	重新发送回复消息的最大尝试次数，因为可恢复错误而失败。
default_reply_timeout = 30	整数值	rpc 回复消息发送的截止时间。
default_rpc_exchange = None	字符串值	RPC 地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_rpc_exchange if set other control_exchange if set else <i>rpc</i>
default_send_timeout = 30	整数值	rpc cast 或调用消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_sender_link_timeout = 600	整数值	调度闲置发送者链接清除的持续时间。过期后分离链接。
group_request_prefix = unicast	字符串值	发送到组中任何服务器时的地址前缀
idle_timeout = 0	整数值	不活跃连接的超时时间（以秒为单位）
link_retry_delay = 10	整数值	重新连接 AMQP 1.0 链接之间的暂停时间，该链接因为可恢复的错误而失败。
multicast_address = multicast	字符串值	在发送异常消息时，附加到地址前缀。消息总线用来识别 fanout 消息。
notify_address_prefix = openstack.org/om/notify	字符串值	所有生成的通知地址的地址前缀
notify_server_credit = 100	整数值	传入通知消息的窗口大小
pre_settled = ['rpc-cast', 'rpc-reply']	多值	发送此类型的消息预先组装。预先设置的消息不会接收来自 peer 的确认信息。请注意：如果发送失败，则预先设置的消息可能会被静默丢弃。允许的值： <i>rpc-call</i> - 发送 RPC Calls <i>pre-settled</i> <i>rpc-reply</i> - 发送 RPC Replies <i>pre-settled</i> <i>rpc-cast</i> - Send RPC Casts <i>pre-settled</i> <i>notify</i> - Send Notifications <i>pre-settled</i>

配置选项 = 默认值	类型	Description
pseudo_vhost = True	布尔值	为不支持虚拟主机的消息总线启用虚拟主机支持（如 qpidd）。当设置为 true 时，虚拟主机名将添加到所有消息总线地址中，从而有效地为每个虚拟主机 <i>创建专用子网</i> 。如果消息总线支持使用 AMQP 1.0 Open performative 中的 <i>hostname</i> 字段作为虚拟主机的名称，设置为 False。
reply_link_credit = 200	整数值	传入 RPC 消息的窗口大小。
rpc_address_prefix = openstack.org/om/rpc	字符串值	所有生成的 RPC 地址的地址前缀
rpc_server_credit = 100	整数值	传入 RPC Request 消息的窗口大小
<code>`sasl_config_dir = `</code>	字符串值	包含 SASL 配置的目录路径
<code>`sasl_config_name = `</code>	字符串值	配置文件名称（不带 .conf 后缀）
<code>`sasl_default_realm = `</code>	字符串值	如果用户名中没有 realm，则要使用的 SASL 域
<code>`sasl_mechanisms = `</code>	字符串值	可接受的 SASL 机制的以空格分隔的列表
server_request_prefix = exclusive	字符串值	发送到特定服务器时使用的地址前缀
ssl = False	布尔值	尝试通过 SSL 连接。如果没有给出其他与 ssl 相关的参数，它将使用系统的 CA-bundle 来验证服务器证书。
<code>`ssl_ca_file = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_cert_file = `</code>	字符串值	自我标识用于客户端身份验证的证书 PEM 文件
<code>`ssl_key_file = `</code>	字符串值	用于为 ssl_cert_file 证书签名的私钥 PEM 文件（可选）
ssl_key_password = None	字符串值	用于解密 ssl_key_file 的密码（如果加密）
ssl_verify_vhost = False	布尔值	默认情况下，SSL 检查服务器证书中的名称与 transport_url 中的主机名匹配。在某些配置中，最好使用虚拟主机名。例如，如果服务器使用 Server Name Indication TLS extension (rfc6066) 来为每个虚拟主机提供证书。如果服务器的 SSL 证书使用虚拟主机名称而不是 DNS 名称，则将 ssl_verify_vhost 设置为 True。

配置选项 = 默认值	类型	Description
trace = False	布尔值	debug : 将 AMQP 帧转储到 stdout
unicast_address = unicast	字符串值	在发送到特定 RPC/Notification 服务器时, 附加到地址前缀。消息总线用来识别发送到单个目的地的消息。

5.1.35. oslo_messaging_kafka

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[oslo_messaging_kafka]` 组下可用的选项。

表 5.34. oslo_messaging_kafka

配置选项 = 默认值	类型	Description
compression_codec = none	字符串值	producer 生成的所有数据的压缩 codec。如果没有设置, 则不会使用压缩。请注意, 这允许的值取决于 kafka 版本
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
consumer_group = oslo_messaging_consumer	字符串值	Kafka 消费者的组 ID。一个组中的消费者将协调消息消耗
enable_auto_commit = False	布尔值	启用异步消费者提交
kafka_consumer_timeout = 1.0	浮点值	Kafka 用户的默认超时
kafka_max_fetch_bytes = 1048576	整数值	Kafka consumer 的最大获取字节数
max_poll_records = 500	整数值	轮询调用中返回的最大记录数
pool_size = 10	整数值	Kafka Consumers 的池大小
producer_batch_size = 16384	整数值	producer async 发送的批处理大小
producer_batch_timeout = 0.0	浮点值	以秒为单位的 KafkaProducer 批处理延迟上限
sasl_mechanism = PLAIN	字符串值	当安全协议是 SASL 时的机制

配置选项 = 默认值	类型	Description
security_protocol = PLAINTEXT	字符串值	用于与代理通信的协议
<code>`ssl_cafile = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_client_cert_file = `</code>	字符串值	用于身份验证的客户端证书 PEM 文件。
<code>`ssl_client_key_file = `</code>	字符串值	用于身份验证的客户端密钥 PEM 文件。
<code>`ssl_client_key_password = `</code>	字符串值	用于身份验证的客户端密钥密码文件。

5.1.36. oslo_messaging_notifications

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[oslo_messaging_notifications]` 组下可用的选项。

表 5.35. oslo_messaging_notifications

配置选项 = 默认值	类型	Description
driver = []	多值	处理发送通知的驱动程序。可能的值有 <code>messaging</code> , <code>messagingv2</code> , <code>routing</code> , <code>log</code> , <code>test</code> , <code>noop</code>
retry = -1	整数值	重新发送因为可恢复错误而发送失败的通知消息的最大尝试次数。0 - 不重试, -1 - indefinite
topics = ['notifications']	列表值	用于 OpenStack 通知的 AMQP 主题。
transport_url = None	字符串值	代表用于通知的消息传递驱动程序的 URL。如果没有设置, 我们回退到用于 RPC 的同一配置。

5.1.37. oslo_messaging_rabbit

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[oslo_messaging_rabbit]` 组下可用的选项。

表 5.36. oslo_messaging_rabbit

配置选项 = 默认值	类型	Description
amqp_auto_delete = False	布尔值	AMQP 中的自动删除队列。
amqp_durable_queues = False	布尔值	在 AMQP 中使用持久队列。

配置选项 = 默认值	类型	Description
direct_mandatory_flag = True	布尔值	(DEPRECATED)为直接发送启用/禁用 RabbitMQ 强制标记。直接发送用作回复，因此在客户端队列不存在时引发 MessageUndeliverable 异常。 MessageUndeliverable 异常将用于循环使发送者恢复。此标志已弃用，且无法停用此功能
enable_cancel_on_failover = False	布尔值	启用 x-cancel-on-ha-failover 标志，使 rabbitmq 服务器将取消并通知 consumerswhen 队列停机
heartbeat_in_pthread = False	布尔值	默认情况下，通过原生 python 线程运行健康检查心跳线程。如果此选项等于 False，则健康检查 heartbeat 将继承父进程的执行模型。例如，如果父进程使用 eventlet/greenlet 对 stdlib 进行补丁，则心跳将通过绿色线程运行。这个选项应该只针对 wsgi 服务设置为 True。
heartbeat_rate = 2	整数值	heartbeat_timeout_threshold 期间的我们检查心跳的频率。
heartbeat_timeout_threshold = 60	整数值	如果 heartbeat 的 keep-alive 失败(0 禁用 heartbeat)，则 Rabbit 代理被视为 down 的秒数。
kombu_compression = None	字符串值	EXPERIMENTAL : Possible 值为 : gzip、bz2。如果没有设置压缩，则不会使用。这个选项可能在以后的版本中不可用。
kombu_failover_strategy = round-robin	字符串值	确定当当前连接到的 RabbitMQ 节点不可用时如何选择下一个 RabbitMQ 节点。仅在配置中提供多个 RabbitMQ 节点时生效。
kombu_missing_consumer_retry_timeout = 60	整数值	在放弃发送其回复前等待缺少的客户端的时间。这个值不应比 rpc_response_timeout 长。
kombu_reconnect_delay = 1.0	浮点值	响应 AMQP 消费者取消通知前等待多久。
rabbit_ha_queues = False	布尔值	尝试在 RabbitMQ 中使用 HA 队列(x-ha-policy: all)。如果更改这个选项，您必须擦除 RabbitMQ 数据库。在 RabbitMQ 3.0 中，在声明队列时，x-ha-policy 参数不再控制队列镜像。如果您只想确保所有队列（带有自动生成的名称的队列除外）都在所有节点上进行镜像，请运行： <code>"rabbitmqctl set_policy HA ^(?!amq\.) gem {"ha-mode": "all"}"</code>
rabbit_interval_max = 30	整数值	RabbitMQ 连接重试的最大间隔。默认值为 30 秒。
rabbit_login_method = AMQPLAIN	字符串值	RabbitMQ 登录方法。

配置选项 = 默认值	类型	Description
rabbit_qos_prefetch_count = 0	整数值	指定要预抓取的消息数量。设置为零可允许无限消息。
rabbit_retry_backoff = 2	整数值	连接到 RabbitMQ 时重试之间的 backoff 时长。
rabbit_retry_interval = 1	整数值	使用 RabbitMQ 重试连接的频率。
rabbit_transient_queues_ttl = 1800	整数值	正整数代表队列 TTL (x-expires) 的持续时间 (以秒为单位)。TTL 期间未使用的队列将被自动删除。参数仅影响 reply 和 fanout 队列。
ssl = False	布尔值	通过 SSL 连接。
<code>`ssl_ca_file = `</code>	字符串值	SSL 证书颁发机构文件 (仅在启用了 SSL 时评估)。
<code>`ssl_cert_file = `</code>	字符串值	SSL 证书文件 (仅在启用了 SSL 时评估)。
<code>`ssl_key_file = `</code>	字符串值	SSL 密钥文件 (仅在启用了 SSL 时 valid)。
<code>`ssl_version = `</code>	字符串值	要使用的 SSL 版本 (仅在启用了 SSL 时评估)。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1_1 和 TLSv1_2 可能在某些发行版中提供。

5.1.38. oslo_middleware

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[oslo_middleware]` 组下可用的选项。

表 5.37. oslo_middleware

配置选项 = 默认值	类型	Description
enable_proxy_headers_parsing = False	布尔值	应用程序是否在代理后面。这决定了中间件是否应该解析标头。

5.1.39. oslo_policy

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[oslo_policy]` 组下可用的选项。

表 5.38. oslo_policy

配置选项 = 默认值	类型	描述
------------	----	----

配置选项 = 默认值	类型	描述
enforce_new_defaults = False	布尔值	这个选项控制在评估策略时是否使用旧的弃用默认值。如果为 True ，则不会评估旧的已弃用的默认值。这意味着，如果旧默认值允许任何现有令牌，但不允许使用新默认值，它将被禁止。我们鼓励启用此标志以及 enforce_scope 标志，以便您可以同时获得新默认值和 scope_type 的好处
enforce_scope = False	布尔值	此选项控制在评估策略时是否强制执行范围。如果为 True ，则请求中使用的令牌范围将与被强制执行策略的 scope_types 进行比较。如果范围不匹配，则引发 InvalidScope 异常。如果为 False ，则会记录一条信息，告知操作员使用了不匹配的范围调用策略。
policy_default_rule = default	字符串值	默认规则。未找到请求的规则时强制执行。
policy_dirs = ['policy.d']	多值	存储策略配置文件的目录。它们可以相对于 <code>config_dir</code> 选项或绝对路径定义的搜索路径中的任何目录。由 <code>policy_file</code> 定义的文件必须存在，才能搜索这些目录。缺少或空目录将被忽略。
policy_file = policy.json	字符串值	将角色映射到给定服务权限的文件的相对或绝对路径。必须使用与配置文件设置相关的相对路径。
remote_content_type = application/x-www-form-urlencoded	字符串值	用于发送和接收基于 REST 策略检查的数据的内容类型
remote_ssl_ca_cert_file = None	字符串值	基于 REST 的策略检查的 ca cert 文件的绝对路径
remote_ssl_client_cert_file = None	字符串值	基于 REST 的策略检查的客户端证书的绝对路径
remote_ssl_client_key_file = None	字符串值	绝对路径客户端密钥文件 REST 基于策略检查
remote_ssl_verify_server_cert = False	布尔值	基于 REST 的策略检查的服务器身份验证

5.1.40. oslo_reports

下表概述了 `/etc/ironic/ironic.conf` 文件中的 **[oslo_reports]** 组下可用的选项。

表 5.39. oslo_reports

配置选项 = 默认值	类型	Description
file_event_handler = None	字符串值	要监视触发报告更改的文件路径，而不是信号。设置此选项会禁用报告的信号触发器。如果应用程序作为 WSGI 应用程序运行，则建议使用此选项而不是信号。
file_event_handler_interval = 1	整数值	设置 file_event_handler 时轮询之间等待的秒数
log_dir = None	字符串值	创建文件的路径

5.1.41. profiler

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[profiler]` 组下可用的选项。

表 5.40. profiler

配置选项 = 默认值	类型	Description
connection_string = messaging://	字符串值	<p>通知程序后端的连接字符串。</p> <p>默认值为 messaging://，它将通知程序设置为 <code>oslo_messaging</code>。</p> <p>可能的值示例：</p> <ul style="list-style-type: none"> ● messaging:// - 使用 <code>oslo_messaging</code> 驱动程序来发送范围。 ● redis://127.0.0.1:6379 - 使用 <code>redis</code> 驱动程序发送 span。 ● MongoDB://127.0.0.1:27017 - 使用 <code>mongodb</code> 驱动程序进行发送范围。 ● Elasticsearch://127.0.0.1:9200 - 使用 <code>elasticsearch</code> 驱动程序来发送 span。 ● Jaeger://127.0.0.1:6831 - 使用 <code>jaeger tracing</code> 作为发送 span 的驱动程序。
enabled = False	布尔值	<p>为此节点上的所有服务启用性能分析。</p> <p>默认值为 <code>False</code>（完全禁用性能分析功能）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● <code>true</code>：启用该功能 ● <code>false</code>：禁用该功能。分析无法通过此项目操作启动。如果性能分析由另一个项目触发，则此项目部分将为空。

配置选项 = 默认值	类型	Description
es_doc_type = notification	字符串值	elasticsearch 中通知索引的文档类型。
es_scroll_size = 10000	整数值	Elasticsearch 批量分割大型请求。此参数定义每个批处理的最大大小（例如：es_scroll_size=10000）。
es_scroll_time = 2m	字符串值	此参数是一个时间值参数（例如：es_scroll_time=2m），指示参与搜索的节点将维护相关资源以继续和支持它。
filter_error_trace = False	布尔值	<p>启用包含 error/exception 的过滤器追踪来分隔的位置。</p> <p>默认值为 False。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用包含 error/exception 的过滤器追踪。 ● false：禁用过滤器。
hmac_keys = SECRET_KEY	字符串值	<p>用于加密上下文数据的 secret 密钥，以用于性能分析。</p> <p>此字符串值应该具有以下格式：<key1>[,<key2>,...<keyn>]，其中每个键都是随机字符串。通过 REST API 触发性能分析的用户必须在 REST API 调用的标头中设置其中一个键，使其包含此节点的性能分析结果。</p> <p>"enabled"标志和"hmac_keys"配置选项都应设置为启用性能分析。此外，为了跨所有服务生成正确的性能分析信息，OpenStack 项目之间至少需要一个关键。这样可确保它可用于客户端生成 trace，其中包含所有可能的资源的信息。</p>
sentinel_service_name = mymaster	字符串值	<p>Redis sentinel 使用服务名称来识别 master redis 服务。此参数定义名称（例如：sentinal_service_name=mymaster）。</p>
socket_timeout = 0.1	浮点值	<p>Redis sentinel 在连接上提供了一个 timeout 选项。此参数定义该超时（例如：socket_timeout=0.1）。</p>

配置选项 = 默认值	类型	Description
trace_sqlalchemy = False	布尔值	<p>在服务中启用 SQL 请求性能分析。</p> <p>默认值为 False (SQL 请求不会被跟踪)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用 SQL 请求性能分析。每个 SQL 查询都将是 trace 的一部分，可以通过花费的时间来分析。 ● false：禁用 SQL 请求性能分析。花费的时间仅显示在更高级别的操作上。无法以这种方式分析单个 SQL 查询。

5.1.42. pxe

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[pxe]` 组下可用的选项。

表 5.41. pxe

配置选项 = 默认值	类型	Description
boot_retry_check_interval = 90	整数值	PXE 引导重试时定期检查间隔（以秒为单位）。如果没有设置 <code>boot_retry_timeout</code> ，则无效。
boot_retry_timeout = None	整数值	应该重试 PXE 引导的超时（以秒为单位）。必须小于 <code>[conductor]deploy_callback_timeout</code> 。默认禁用此选项。
default_ephemeral_format = ext4	字符串值	临时分区的默认文件系统格式（如果创建）。
dir_permission = None	整数值	创建时将应用到 TFTP 文件夹的权限。这应该设置为权限，以便 <code>tftpserver</code> 有权访问配置的 TFTP 文件夹的内容。只有在操作系统的 <code>umask</code> 的限制限制时才需要此设置，以便 <code>ironic-conductor</code> 创建无法由 TFTP 服务器读取的文件。设置为 <code><None></code> 将导致操作系统的 <code>umask</code> 用于创建新的 <code>tftp</code> 文件夹。建议指定八进制表示。例如： <code>0o755</code>
enable_netboot_fallback = False	布尔值	如果为 <code>True</code> ，即使使用本地引导的节点也会生成 PXE 环境。当驱动程序无法将节点切换到本地引导时，这很有用，例如，在无法进行持久性引导的机器上使用 SNMP 或 Redfish。主要用于独立的 <code>ironic</code> ，因为 <code>Neutron</code> 会阻止不正确的 PXE 引导。

配置选项 = 默认值	类型	Description
image_cache_size = 20480	整数值	master 镜像的最大缓存大小（单位 MiB），包括正在使用的镜像。
image_cache_ttl = 10080	整数值	缓存中旧 master 镜像的最大 TTL（以分钟为单位）。
images_path = /var/lib/ironic/images/	字符串值	在 ironic-conductor 节点上，镜像存储在磁盘上的目录。
instance_master_path = /var/lib/ironic/master_images	字符串值	在 ironic-conductor 节点上，master 实例镜像存储在磁盘上的目录。设置为空字符串可禁用镜像缓存。
ip_version = 4	字符串值	用于 PXE 引导的 IP 版本。默认值为 4。实验性
ipxe_boot_script = \$pybasedir/drivers/modules/boot.ipxe	字符串值	在 ironic-conductor 节点上，主 iPXE 脚本文件的路径。
ipxe_bootfile_name = undionly.kpxe	字符串值	Bootfile DHCP 参数。
ipxe_bootfile_name_by_arch = {}	dict 值	每个节点架构的 Bootfile DHCP 参数。例如： aarch64:ipxe_aa64.efi
ipxe_config_template = \$pybasedir/drivers/modules/ipxe_config.template	字符串值	在 ironic-conductor 节点上，用于 iPXE 操作的模板文件。
ipxe_timeout = 0	整数值	通过 iPXE 下载镜像的超时值（以秒为单位）。默认值为 0（无超时）
ipxe_use_swift = False	布尔值	使用临时 URL 从 swift 下载部署和救援镜像。如果设置为 false（默认），镜像将下载到 ironic-conductor 节点，并通过其本地 HTTP 服务器提供。仅在使用 ipxe 兼容引导接口时适用。
pxe_append_params = nofb nomodeset vga=normal	字符串值	为 baremetal PXE 引导附加参数。
pxe_bootfile_name = pxelinux.0	字符串值	Bootfile DHCP 参数。
pxe_bootfile_name_by_arch = {}	dict 值	每个节点架构的 Bootfile DHCP 参数。例如： aarch64:grubaa64.efi

配置选项 = 默认值	类型	Description
pxe_config_subdir = pxelinux.cfg	字符串值	创建代表节点上端口 MAC 或 IP 地址的符号链接的目录，并允许引导装载程序为节点加载 PXE 文件。此目录名称与 PXE 或 iPXE 文件夹相对。
pxe_config_template = \$pybasedir/drivers/modules/pxe_config.template	字符串值	在 ironic-conductor 节点上，用于 PXE 加载程序配置的模板文件。
pxe_config_template_by_arch = {}	dict 值	在 ironic-conductor 节点上，用于每个节点的架构 PXE 配置的模板文件。例如： aarch64:/opt/share/grubaa64_pxe_config.template
tftp_master_path = /tftpboot/master_images	字符串值	在 ironic-conductor 节点上，master TFTP 镜像存储在磁盘上的目录。设置为空字符串可禁用镜像缓存。
tftp_root = /tftpboot	字符串值	ironic-conductor 节点的 TFTP 根路径。ironic-conductor 必须具有对此路径的读/写权限。
tftp_server = \$my_ip	字符串值	ironic-conductor 节点的 TFTP 服务器的 IP 地址。
uefi_ipxe_bootfile_name = ipxe.efi	字符串值	UEFI 引导模式的 Bootfile DHCP 参数。如果您在使用引导时遇到问题，请尝试 snponly.efi。
uefi_pxe_bootfile_name = bootx64.efi	字符串值	UEFI 引导模式的 Bootfile DHCP 参数。
uefi_pxe_config_template = \$pybasedir/drivers/modules/pxe_grub_config.template	字符串值	在 ironic-conductor 节点上，UEFI 引导装载程序的 PXE 配置的模板文件。这通常用于 GRUB 特定模板。

5.1.43. Redfish

下表概述了 `/etc/ironic/ironic.conf` 文件中的 **[redfish]** 组下可用的选项。

表 5.42. Redfish

配置选项 = 默认值	类型	描述
auth_type = auto	字符串值	Redfish HTTP 客户端验证方法。
connection_attempts = 5	整数值	尝试连接到 Redfish 的最大尝试次数

配置选项 = 默认值	类型	描述
connection_cache_size = 1000	整数值	最大 Redfish 客户端连接缓存大小。Redfish 驱动程序会努力重复使用经过身份验证的 BMC 连接（通过 Redfish Session Service 获得）。这个选项限制了要维护的最大连接数。 0 代表完全禁用客户端连接缓存。
connection_retry_interval = 4	整数值	尝试连接到 Redfish 间等待的秒数
file_permission = 420	整数值	swift-less 镜像的文件权限，使用文件访问权限的八进制权限表示。此设置默认为 644 ，或 Python 中的八进制数 0o644 。此设置必须设置为八进制数表示，即以 0o 开始。
firmware_update_fail_interval = 60	整数值	检查失败的固件更新任务之间等待的秒数
firmware_update_status_interval = 60	整数值	检查完成的固件更新任务之间等待的秒数
kernel_append_params = nofb nomodeset vga=normal	字符串值	传递给实例内核的其他内核参数。这些参数可由内核或应用程序使用，方法是读取 <code>/proc/cmdline</code> 。严重的 <code>cmdline</code> 大小限制！可以被 instance_info/kernel_append_params 属性覆盖。
raid_config_fail_interval = 60	整数值	检查失败的 raid 配置任务之间等待的秒数
raid_config_status_interval = 60	整数值	检查完成的 raid 配置任务之间等待的秒数
swift_container = ironic_redfish_container	字符串值	用于存储 Redfish 驱动程序数据的 Swift 容器。仅在启用 use_swift 时应用。
swift_object_expiry_timeout = 900	整数值	Swift 对象自动过期的时间（以秒为单位）。仅在启用 use_swift 时应用。
use_swift = True	布尔值	将为虚拟介质引导生成的 ISO 镜像上传到 Swift，然后将临时 URL 传递给 BMC 以引导节点。如果设置为 <code>false</code> ，则镜像放置在 <code>ironic-conductor</code> 节点上，并通过其本地 HTTP 服务器提供。

5.1.44. service_catalog

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[service_catalog]` 组下可用的选项。

表 5.43. service_catalog

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
connect-retries = None	整数值	应该尝试连接错误的最大重试次数。
connect-retry-delay = None	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围的域名
endpoint-override = None	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version , min-version , 和/或 max-version 选项。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
max-version = None	字符串值	给定 API 的最大主要版本，旨在作为带有 min_version 的范围的上限。与版本相互排斥。
min-version = None	字符串值	给定 API 的最低主要版本，旨在作为 max_version 的范围低限。与版本相互排斥。如果给出了 min_version，且没有 max_version，它就如同 max version 为 "latest"。
password = None	字符串值	用户的密码

配置选项 = 默认值	类型	Description
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = baremetal	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。

配置选项 = 默认值	类型	Description
version = None	字符串值	给定主版本 API 版本内用于端点 URL 发现的最低 Major API 版本。与 min_version 和 max_version 相互排斥

5.1.45. snmp

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[snmp]` 组下可用的选项。

表 5.44. snmp

配置选项 = 默认值	类型	Description
power_timeout = 10	整数值	等待电源操作完成的秒数
reboot_delay = 0	整数值	重启（关闭和再次打开）之间休眠的时间（以秒为单位）
udp_transport_retries = 5	整数值	最多 UDP 请求重试次数，0 代表没有重试。
udp_transport_timeout = 1.0	浮点值	用于 UDP 传输的响应超时（以秒为单位）。超时应该是 0.5 秒的倍数，并适用于每个重试。

5.1.46. ssl

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[ssl]` 组下可用的选项。

表 5.45. ssl

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	用于验证连接客户端的 CA 证书文件。
cert_file = None	字符串值	安全启动服务器时要使用的证书文件。
密码 = None	字符串值	设置可用密码列表。值应该是 OpenSSL 密码列表格式的字符串。
key_file = None	字符串值	安全启动服务器时使用的私钥文件。
version = None	字符串值	要使用的 SSL 版本（仅在启用了 SSL 时评估）。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1_1 和 TLSv1_2 可能在某些发行版中提供。

5.1.47. swift

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[swift]` 组下可用的选项。

表 5.46. swift

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
connect-retries = None	整数值	应该尝试连接错误的最大重试次数。
connect-retry-delay = None	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围的域名
endpoint-override = None	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version , min-version , 和/或 max-version 选项。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
max-version = None	字符串值	给定 API 的最大主要版本，旨在作为带有 min_version 的范围的上限。与版本相互排斥。
min-version = None	字符串值	给定 API 的最低主要版本，旨在作为 max_version 的范围低限。与版本相互排斥。如果给出了 min_version，且没有 max_version，它就如同 max version 为 "latest"。
password = None	字符串值	用户的密码

配置选项 = 默认值	类型	Description
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = object-store	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
swift_max_retries = 2	整数值	失败前重试 Swift 请求的次数上限。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名

配置选项 = 默认值	类型	Description
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。
version = None	字符串值	给定主版本 API 版本内用于端点 URL 发现的最低 Major API 版本。与 min_version 和 max_version 相互排斥

5.1.48. xclarity

下表概述了 `/etc/ironic/ironic.conf` 文件中的 `[xclarity]` 组下可用的选项。

表 5.47. xclarity

配置选项 = 默认值	类型	Description
manager_ip = None	字符串值	XClarity Controller 的 IP 地址。这里的配置已弃用，并将在 Stein 发行版本中删除。请更新 driver_info 字段以使用 "xclarity_manager_ip" 替代
password = None	字符串值	XClarity Controller 用户名的密码。这里的配置已弃用，并将在 Stein 发行版本中删除。请更新 driver_info 字段以使用 "xclarity_password" 替代
port = 443	端口值	用于 XClarity Controller 连接的端口。
username = None	字符串值	XClarity Controller 的用户名。这里的配置已弃用，并将在 Stein 发行版本中删除。请更新 driver_info 字段以使用 "xclarity_username" 替代

第 6 章 IRONIC-INSPECTOR

下面的章节包含有关 `ironic-inspector` 服务中的配置选项的信息。

6.1. INSPECTOR.CONF

本节包含 `/etc/ironic-inspector/inspector.conf` 文件的选项。

6.1.1. DEFAULT

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
<code>api_max_limit = 1000</code>	整数值	限制 API list-call 返回的元素数量
<code>auth_strategy = keystone</code>	字符串值	ironic-inspector API 上使用的身份验证方法。 "noauth"、"keystone" 或 "http_basic" 是有效的选项。 "noauth" 将禁用所有身份验证。
<code>can_manage_boot = True</code>	布尔值	ironic-inspector 的当前安装是否可以管理节点的 PXE 引导。如果设置为 False，API 将拒绝缺少 <code>manage_boot</code> 的内省请求或设置为 True。
<code>clean_up_period = 60</code>	整数值	重复清理节点和旧节点状态信息的时间（以秒为单位）。警告：如果设为 0，则定期任务被禁用，并且检查器不会与 ironic 同步以完成内部清理过程。如果部署使用了 PXE 过滤器，并不建议导致 ironic-inspector 到期定期清理活动。
<code>debug = False</code>	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。
<code>default_log_levels = ['sqlalchemy=WARNING', 'iso8601=WARNING', 'requests=WARNING', 'urllib3.connectionpool=WARNING', 'keystonemiddleware=WARNING', 'keystoneauth=WARNING', 'ironicclient=WARNING', 'amqp=WARNING', 'amqplib=WARNING', 'oslo.messaging=WARNING', 'oslo_messaging=WARNING']</code>	列表值	以 <code>logger=LEVEL</code> 的形式列出软件包日志级别列表。如果设置了 <code>log_config_append</code> ，则忽略此选项。

配置选项 = 默认值	类型	Description
enable_mdns = False	布尔值	是否启用通过多播 DNS 发布 ironic-inspector API 端点。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
Host = <based on operating system>	字符串值	此节点的名称。这可以是不透明标识符。它不一定是主机名、FQDN 或 IP 地址。但是，节点名称必须在 AMQP 密钥内有效，如果使用 ZeroMQ、有效的主机名、FQDN 或 IP 地址。
http_basic_auth_user_file = /etc/ironic-inspector/htpasswd	字符串值	auth_strategy=http_basic 时使用的 Apache 格式用户身份验证文件的路径
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。
introspection_delay = 5	整数值	两个内省之间的延迟（以秒为单位）。仅在引导由 ironic-inspector（例如 manage_boot==True）管理时才适用。
ipmi_address_fields = ['redfish_address', 'ilo_address', 'drac_host', 'drac_address', 'ibmc_address']	列表值	等同于 ipmi_address 的 ironic driver_info 字段。
leader_election_interval = 10	整数值	领导选举机制之间的间隔（以秒为单位）。
listen_address = ::	字符串值	要侦听的 IP。
listen_port = 5050	端口值	要侦听的端口。
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 log-date-format）。

配置选项 = 默认值	类型	Description
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	(可选) 用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	(可选) 要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	字符串值	使用上下文格式化用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_debug_format_sufffix = %(funcName)s %(pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 <code>DEBUG</code> 时，要附加到日志消息的其他数据。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_default_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s	字符串值	上下文未定义时用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 <code>oslo_log.formatters.ContextFormatter</code> 使用

配置选项 = 默认值	类型	Description
logging_user_identity_format = %(user) s %(tenant) s %(domain) s %(user_domain) s %(project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_concurrency = 1000	整数值	绿色线程池大小。
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size", 则忽略这个选项。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
rootwrap_config = /etc/ironic-inspector/rootwrap.conf	字符串值	用于以 root 用户身份运行命令的 rootwrap 配置文件的路径
standalone = True	布尔值	是否作为独立服务运行 ironic-inspector。它是 EXPERIMENTAL，设置为 False。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
timeout = 3600	整数值	内省被视为失败的超时，设置为 0 以禁用。
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。

配置选项 = 默认值	类型	Description
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_ssl = False	布尔值	SSL Enabled/Disabled
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 log_file 选项并使用 Linux 平台时，才有意义。如果设置了 log_config_append，则忽略此选项。

6.1.2. 功能

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 **[capabilities]** 组下可用的选项。

表 6.1. 功能

配置选项 = 默认值	类型	Description
boot_mode = False	布尔值	是否存储引导模式(BIOS 或 UEFI)。
cpu_flags = {'aes': 'cpu_aes', 'pdpe1gb': 'cpu_hugepages_1g', 'pse': 'cpu_hugepages', 'smx': 'cpu_txt', 'svm': 'cpu_vt', 'vmx': 'cpu_vt'}	dict 值	如果存在此标志，在 CPU 标记和能力之间进行映射。

6.1.3. 协调

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 **[coordination]** 组下可用的选项。

表 6.2. 协调

配置选项 = 默认值	类型	Description
backend_url = memcached://localhost:11211	字符串值	用于分布式协调的后端 URL。实验性。

6.1.4. CORS

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[cors]` 组下可用的选项。

表 6.3. CORS

配置选项 = 默认值	类型	Description
<code>allow_credentials = True</code>	布尔值	指明实际请求可以包含用户凭证
<code>allow_headers = ['X-Auth-Token', 'X-OpenStack-Ironic-Inspector-API-Minimum-Version', 'X-OpenStack-Ironic-Inspector-API-Maximum-Version', 'X-OpenStack-Ironic-Inspector-API-Version']</code>	列表值	指定在实际请求过程中可以使用哪些标头字段名称。
<code>allow_methods = ['GET', 'POST', 'PUT', 'HEAD', 'PATCH', 'DELETE', 'OPTIONS']</code>	列表值	指明在实际请求期间可以使用哪些方法。
<code>allowed_origin = None</code>	列表值	指明此资源是否可以与请求"origin"标头中收到的域共享。格式： <code>"<protocol>://<host>[:<port>]"</code> ，没有尾部斜杠。示例： https://horizon.example.com
<code>expose_headers = []</code>	列表值	指明哪些标头可以安全地公开给 API。默认为 HTTP Simple Headers。
<code>max_age = 3600</code>	整数值	CORS preflight 请求的最大缓存期限。

6.1.5. database

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[database]` 组下可用的选项。

表 6.4. database

配置选项 = 默认值	类型	Description
<code>backend = sqlalchemy</code>	字符串值	用于数据库的后端。
<code>connection = None</code>	字符串值	用于连接到数据库的 SQLAlchemy 连接字符串。
<code>connection_debug = 0</code>	整数值	SQL 调试信息的详细程度：0=None, 100=Everything。
<code>connection_parameters = `</code>	字符串值	用于连接时附加到连接 URL 的可选 URL 参数；指定 <code>param1=value1&param2=value2&...</code>

配置选项 = 默认值	类型	Description
connection_recycle_time = 3600	整数值	连接池中存在的连接的时间超过这个秒数，下一次从池中签出它们的时间将替换为新的连接。
connection_trace = False	布尔值	将 Python 堆栈跟踪作为注释字符串添加到 SQL。
db_inc_retry_interval = True	布尔值	如果为 True，将数据库操作的重试间隔增加到 db_max_retry_interval。
db_max_retries = 20	整数值	在引发错误前出现连接错误或死锁错误时的最大重试次数。设置为 -1 以指定无限重试计数。
db_max_retry_interval = 10	整数值	如果设置了 db_inc_retry_interval，则数据库操作重试之间的最大秒数。
db_retry_interval = 1	整数值	数据库事务重试之间的秒数。
max_overflow = 50	整数值	如果设置，请将这个值用于 SQLAlchemy 的 max_overflow。
max_pool_size = 5	整数值	在池中保留的最大 SQL 连接数。将值设为 0 表示没有限制。
max_retries = 10	整数值	启动期间数据库连接重试的最大数量。设置为 -1 以指定无限重试计数。
mysql_enable_ndb = False	布尔值	如果为 True，则透明地启用对处理 MySQL 集群 (NDB) 的支持。
mysql_sql_mode = TRADITIONAL	字符串值	用于 MySQL 会话的 SQL 模式。这个选项（包括默认选项）会覆盖任何 server-set SQL 模式。要使用服务器配置设置的任何 SQL 模式，请将其设置为 no 值。 示例：mysql_sql_mode=
pool_timeout = None	整数值	如果设置，请将这个值用于 SQLAlchemy 的 pool_timeout。
retry_interval = 10	整数值	重新尝试打开 SQL 连接之间的间隔。
slave_connection = None	字符串值	用于连接到从数据库的 SQLAlchemy 连接字符串。
sqlite_synchronous = True	布尔值	如果为 True，则 SQLite 使用同步模式。
use_db_reconnect = False	布尔值	启用在连接丢失时进行数据库重新连接的实验性使用。

6.1.6. discovery

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[discovery]` 组下可用的选项。

表 6.5. discovery

配置选项 = 默认值	类型	Description
<code>enabled_bmc_address_version = ['4', '6']</code>	列表值	在 Ironic 中注册新节点时使用的 BMC 地址的 IP 版本。默认为 "4,6"。可以是 "4"（仅限使用 v4 地址）、"4,6"（v4 地址具有更高的优先级，如果两者都存在，则会忽略 v6），"6,4"（v6 有更高的优先级，对于使用 v4 地址的 BMC 则会使用 v4 地址，这与 "4,6" 相反）、"6"（仅使用 v6 地址并忽略 v4 版本）。
<code>enroll_node_driver = fake-hardware</code>	字符串值	在 Ironic 中创建新节点时注册 hook 使用的 Ironic 驱动程序名称。
<code>enroll_node_fields = {}</code>	dict 值	在新发现的节点上要设置的其他字段。

6.1.7. dnsmasq_pxe_filter

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[dnsmasq_pxe_filter]` 组下可用的选项。

表 6.6. dnsmasq_pxe_filter

配置选项 = 默认值	类型	Description
<code>dhcp_hostsdire = /var/lib/ironic-inspector/dhcp-hostsdir</code>	字符串值	公开给 dnsmasq 的 MAC 地址缓存目录。该目录应该完全完全控制驱动程序。
<code>dnsmasq_start_command = `</code>	字符串值	在过滤初始化时启动 dnsmasq 服务(shell)命令行。默认：不启动。
<code>dnsmasq_stop_command = `</code>	字符串值	在检查器（错误）退出时，用于停止 dnsmasq 服务的 (shell)命令行。默认：不停止。
<code>purge_dhcp_hostsdire = True</code>	布尔值	在驱动程序初始化后清除 hostsdir。只有在部署检查器时，才会执行设置为 false，以便在同一主机和命名空间中执行多个进程。在这种情况下，Operator 负责设置自定义清理功能。

6.1.8. extra_hardware

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[extra_hardware]` 组下可用的选项。

表 6.7. extra_hardware

配置选项 = 默认值	类型	描述
strict = False	布尔值	如果为 True，如果至少有一个记录太短，则拒绝解析额外的数据。另外，即使解析失败，删除传入的 "data"。

6.1.9. iptables

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[iptables]` 组下可用的选项。

表 6.8. iptables

配置选项 = 默认值	类型	Description
dnsmasq_interface = br-ctlplane	字符串值	dnsmasq 侦听的接口，默认为虚拟机的。
ethoib_interfaces = []	列表值	用于物理访问 DHCP 网络的 Inspector 主机上的以太网 Over InfiniBand 接口列表。多个接口将附加到 <code>dnsmasq_interface</code> 中指定的绑定或网桥。根据这些接口上邻居 MAC 列表，InfiniBand 节点的 MAC 会列入黑名单。
firewall_chain = ironic-inspector	字符串值	要使用的 iptables 链名称。
ip_version = 4	字符串值	用于 iptables 过滤器的 IP 版本。默认值为 4。

6.1.10. ironic

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[ironic]` 组下可用的选项。

表 6.9. ironic

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
connect-retries = None	整数值	应该尝试连接错误的最大重试次数。

配置选项 = 默认值	类型	Description
connect-retry-delay = None	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
endpoint-override = None	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version , min-version , 和/或 max-version 选项。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
max-version = None	字符串值	给定 API 的最大主要版本，旨在作为带有 min_version 的范围的上限。与版本相互排斥。
max_retries = 30	整数值	冲突错误(HTTP 409)时的最大重试次数。
min-version = None	字符串值	给定 API 的最低主要版本，旨在作为 max_version 的范围低限。与版本相互排斥。如果给出了 min_version，且没有 max_version，它就如同 max version 为 "latest"。
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。

配置选项 = 默认值	类型	Description
retry_interval = 2	整数值	冲突错误(HTTP 409)的重试间隔。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = baremetal	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。
version = None	字符串值	给定主版本 API 版本内用于端点 URL 发现的最低 Major API 版本。与 min_version 和 max_version 相互排斥

6.1.11. keystone_authtoken

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 **[keystone_authtoken]** 组下可用的选项。

表 6.10. keystone_authtoken

配置选项 = 默认值	类型	Description
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
auth_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点以进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。 这个选项已弃用，现在使用 <code>www_authenticate_uri</code>，并将在 S 发行版本中删除。弃用自：Queens *reason:* auth_uri 选项已弃用，现在使用 <code>www_authenticate_uri</code>，并将在 S 发行版本中删除。
auth_version = None	字符串值	Identity API 端点的 API 版本。
cache = None	字符串值	存储 Swift 缓存对象的请求环境密钥。当使用 Swift 缓存部署 auth_token 中间件时，请使用此选项使中间件共享具有 swift 的缓存后端。否则，使用 memcached_servers 选项。
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。默认为系统 CA。
certFile = None	字符串值	如果身份服务器需要客户端证书，则需要此项
delay_auth_decision = False	布尔值	不要在中间件内处理授权请求，而是将授权决策委派给下游 WSGI 组件。
enforce_token_bind = permissive	字符串值	用于控制令牌绑定的使用和类型。可以设置为 "disabled" 以不检查令牌绑定。如果绑定类型是已知服务器的绑定类型，则验证绑定信息，并忽略它（如 "permissive"），但如果绑定类型未知，则令牌将被拒绝。"必需"令牌绑定需要允许任何形式的令牌绑定。最后，令牌中必须存在的绑定方法名称。
http_connect_timeout = None	整数值	请求超时值与身份 API 服务器通信。
http_request_max_retries = 3	整数值	与身份 API 服务器通信时，我们尝试重新连接的次数。

配置选项 = 默认值	类型	Description
include_service_catalog = True	布尔值	(可选) 指示是否设置 X-Service-Catalog 标头。如果为 False, 则中间件不会要求在令牌验证时请求服务目录, 且不会设置 X-Service-Catalog 标头。
insecure = False	布尔值	验证 HTTPS 连接。
interface = internal	字符串值	用于 Identity API 端点的接口。有效值为 "public", "internal" (默认) 或 "admin"。
keyfile = None	字符串值	如果身份服务器需要客户端证书, 则需要此项
memcache_pool_connection_timeout = 10	整数值	(可选) 操作等待从池中获取 memcached 客户端连接的秒数。
memcache_pool_dead_retry = 300	整数值	(可选) 在再次尝试前 memcached 服务器被视为 dead 的秒数。
memcache_pool_maxsize = 10	整数值	(可选) 到每个 memcached 服务器的最大打开连接数。
memcache_pool_socket_timeout = 3	整数值	(可选) 与 memcached 服务器通信的套接字超时 (以秒为单位)。
memcache_pool_unused_timeout = 60	整数值	(可选) 与 memcached 的连接在池关闭前未使用的秒数。
memcache_secret_key = None	字符串值	(可选, 如果定义了 memcache_security_strategy) 此字符串用于密钥派生。
memcache_security_strategy = None	字符串值	(可选) 如果定义, 则指示令牌数据是否应进行身份验证或加密。如果 MAC, 令牌数据在缓存中进行身份验证 (使用 HMAC)。如果 ENCRYPT, 令牌数据会在缓存中加密并进行身份验证。如果该值不是这些选项之一或为空, 则 auth_token 会在初始化时引发异常。
memcache_use_advanced_pool = False	布尔值	(可选) 使用 advanced (事件安全) memcached 客户端池。高级池只能在 python 2.x 下工作。
memcached_servers = None	列表值	(可选) 指定要用于缓存的 memcached 服务器列表。如果未定义, 令牌将会被缓存在进程中。
region_name = None	字符串值	找到身份服务器的区域。

配置选项 = 默认值	类型	Description
service_token_roles = ['service']	列表值	需要出现在服务令牌中的角色。服务令牌可以请求可以使用过期的令牌，因此此检查应严格控制只有实际服务应发送此令牌。此处的角色作为 ANY 检查应用，因此此列表中的任何角色都必须存在。为了向后兼容的原因，这目前只会影响 allow_expired 检查。
service_token_roles_required = False	布尔值	为了向后兼容的原因，我们必须允许有效的服务令牌传递，但不允许将 service_token_roles 检查视为有效。在以后的发行版本中设置此 true 将成为默认设置，并应启用（如果可能）。
service_type = None	字符串值	服务的名称或类型（在服务目录中出现）。这用于验证具有受限访问规则的令牌。
token_cache_time = 300	整数值	为了防止过度努力验证令牌，中间件会缓存之前可预见的令牌（以秒为单位）。设置为 -1 可完全禁用缓存。
www_authenticate_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。

6.1.12. oslo_policy

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[oslo_policy]` 组下可用的选项。

表 6.11. oslo_policy

配置选项 = 默认值	类型	描述
enforce_new_defaults = False	布尔值	这个选项控制在评估策略时是否使用旧的弃用默认值。如果为 True ，则不会评估旧的已弃用的默认值。这意味着，如果旧默认值允许任何现有令牌，但不允许使用新默认值，它将被禁止。我们鼓励启用此标志以及 enforce_scope 标志，以便您可以同时获得新默认值和 scope_type 的好处
enforce_scope = False	布尔值	此选项控制在评估策略时是否强制执行范围。如果为 True ，则请求中使用的令牌范围将与被强制执行策略的 scope_types 进行比较。如果范围不匹配，则引发 InvalidScope 异常。如果为 False ，则会记录一条信息，告知操作员使用了不匹配的范围调用策略。

配置选项 = 默认值	类型	描述
policy_default_rule = default	字符串值	默认规则。未找到请求的规则时强制执行。
policy_dirs = ['policy.d']	多值	存储策略配置文件的目录。它们可以相对于 config_dir 选项或绝对路径定义的搜索路径中的任何目录。由 policy_file 定义的文件必须存在，才能搜索这些目录。缺少或空目录将被忽略。
policy_file = policy.json	字符串值	将角色映射到给定服务权限的文件的相对或绝对路径。必须使用与配置文件设置相关的相对路径。
remote_content_type = application/x-www-form-urlencoded	字符串值	用于发送和接收基于 REST 策略检查的数据的内容类型
remote_ssl_ca_cert_file = None	字符串值	基于 REST 的策略检查的 ca cert 文件的绝对路径
remote_ssl_client_cert_file = None	字符串值	基于 REST 的策略检查的客户端证书的绝对路径
remote_ssl_client_key_file = None	字符串值	绝对路径客户端密钥文件 REST 基于策略检查
remote_ssl_verify_server_cert = False	布尔值	基于 REST 的策略检查的服务器身份验证

6.1.13. pci_devices

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[pci_devices]` 组下可用的选项。

表 6.12. pci_devices

配置选项 = 默认值	类型	Description
alias = []	多值	由 <code>vendor_id</code> 和 <code>product_id</code> 字段标识的 PCI 设备的别名。Format: <code>{"vendor_id": "1234", "product_id": "5678", "name": "pci_dev1"}</code>

6.1.14. port_physnet

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[port_physnet]` 组下可用的选项。

表 6.13. port_physnet

配置选项 = 默认值	类型	描述
cidr_map = []	列表值	IP 子网 CIDR 到物理网络的映射。当启用了 <code>physnet_cidr_map</code> 处理 hook 时，baremetal 端口的 <code>physical_network</code> 属性会根据这个映射填充。

6.1.15. processing

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[processing]` 组下可用的选项。

表 6.14. processing

配置选项 = 默认值	类型	Description
add_ports = pxe	字符串值	在内省期间作为端口添加的 MAC 地址。可能的值： all（所有 MAC 地址）、Active（带有 IP 地址的 NIC 的 MAC 地址）、pxe（仅 NIC 节点 PXE 引导的 MAC 地址，如果 ramdisk 没有提供 PXE MAC，则回退到“主动”）。
always_store_ramdisk_logs = False	布尔值	是否存储 ramdisk 日志，即使它没有返回错误消息（取决于所设置的“ramdisk_logs_dir”选项）。
default_processing_hooks = ramdisk_error,root_disk_selection,scheduler,validate_interfaces,capabilities,pci_devices	字符串值	用于处理管道的默认 hook 的逗号分隔列表。hook 调度程序使用 Nova 调度程序所需的最小属性更新节点。hook <code>validate_interfaces</code> 确保 ramdisk 提供有效的 NIC 数据。除非真正知道您的操作，否则请不要排除这两个操作。
disk_partitioning_spacing = True	布尔值	对于分区，是否保留 1 GiB 的磁盘大小。只有在与 IPA 用作 ramdisk 时，只有在 ramdisk 端计算旧的 ramdisk <code>local_gb</code> 时生效。
keep_ports = all	字符串值	在内省后保留哪些端口（已在节点上就绪）。可能的值：所有（请勿删除任何内容）、存在（保持内省数据中存在 MAC 的端口），添加（只维护内省期间添加的 MAC）。
node_not_found_hook = None	字符串值	当检查器从尚未了解的节点接收检查信息时，要运行的 hook 名称。默认会忽略此 hook。
overwrite_existing = True	布尔值	是否覆盖节点数据库中的现有值。禁用此选项以使内省非破坏性操作。
permit_active_introspection = False	布尔值	是否处理处于 running 状态的节点。

配置选项 = 默认值	类型	Description
power_off = True	布尔值	如果通过 <i>permit_active_introspection</i> 配置选项启用了该功能，则是否在 active.Nodes 或 rescue 状态后关闭节点。
processing_hooks = \$default_processing_hooks	字符串值	以逗号分隔的用于处理管道的 hook 列表。这样做的默认值为 <i>\$default_processing_hooks</i> ，可以在默认值之前或之后添加 hook，如 "prehook,\$default_processing_hooks,posthook"。
ramdisk_logs_dir = None	字符串值	如果设置，来自 ramdisk 的日志将存储在这个目录中。
ramdisk_logs_filename_format = {uuid}_{dt:%Y%m%d-%H%M%S.%f}.tar.gz	字符串值	用于存储 ramdisk 日志的文件名模板。可以使用以下替换：{uuid} - 节点 UUID 或 "unknown"，{bmc} - 节点 BMC 地址或 "unknown"，{dt} - 当前 UTC 日期和时间 {mac} - PXE 引导 MAC 或 "unknown"。
store_data = none	字符串值	用于存储内省数据的存储后端。可能的值有： <i>none</i> 、 <i>database</i> 和 <i>swift</i> 。如果设置为 <i>none</i> ，则不会存储内省数据。
update_pxe_enabled = True	布尔值	是否根据内省数据更新 pxe_enabled 值。如果 [processing]overwrite_existing 设为 False，则此选项无效

6.1.16. pxe_filter

下表概述了 */etc/ironic-inspector/inspector.conf* 文件中的 **[pxe_filter]** 组下可用的选项。

表 6.15. pxe_filter

配置选项 = 默认值	类型	描述
deny_unknown_macs = False	布尔值	默认情况下，检查器将在内省处于活动状态时为任何节点打开 DHCP 服务器。当内省处于活跃状态时，为未知 MAC 地址打开 DHCP，供用户添加没有端口的节点到 ironic，并且具有基于节点内省结果的 ironic-inspector 注册端口。注意：如果此选项为 True，则节点在内省前必须至少有一个注册的端口。
driver = iptables	字符串值	要使用的 PXE 引导过滤器驱动程序，可能过滤器有："iptables"、"dnsmasq"和"noop"。设置 "noop" 可禁用防火墙过滤。
sync_period = 15	整数值	重复过滤器定期更新的时间（以秒为单位）。

6.1.17. service_catalog

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[service_catalog]` 组下可用的选项。

表 6.16. service_catalog

配置选项 = 默认值	类型	Description
<code>auth-url = None</code>	字符串值	身份验证 URL
<code>auth_type = None</code>	字符串值	要加载的身份验证类型
<code>cafile = None</code>	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
<code>certFile = None</code>	字符串值	PEM 编码的客户端证书文件
<code>collect-timing = False</code>	布尔值	收集每个 API 调用计时信息。
<code>connect-retries = None</code>	整数值	应该尝试连接错误的最大重试次数。
<code>connect-retry-delay = None</code>	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
<code>default-domain-id = None</code>	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
<code>default-domain-name = None</code>	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
<code>domain-id = None</code>	字符串值	范围的域 ID
<code>domain-name = None</code>	字符串值	范围的域名
<code>endpoint-override = None</code>	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 <code>version</code> , <code>min-version</code> , 和/或 <code>max-version</code> 选项。
<code>insecure = False</code>	布尔值	验证 HTTPS 连接。
<code>keyfile = None</code>	字符串值	PEM 编码的客户端证书密钥文件
<code>max-version = None</code>	字符串值	给定 API 的最大主要版本，旨在作为带有 <code>min_version</code> 的范围的上限。与版本相互排斥。
<code>min-version = None</code>	字符串值	给定 API 的最低主要版本，旨在作为 <code>max_version</code> 的范围低限。与版本相互排斥。如果给出了 <code>min_version</code> ，且没有 <code>max_version</code> ，它就如同 <code>max version</code> 为 "latest"。

配置选项 = 默认值	类型	Description
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = baremetal-introspection	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名

配置选项 = 默认值	类型	Description
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。
version = None	字符串值	给定主版本 API 版本内用于端点 URL 发现的最低 Major API 版本。与 min_version 和 max_version 相互排斥

6.1.18. swift

下表概述了 `/etc/ironic-inspector/inspector.conf` 文件中的 `[swift]` 组下可用的选项。

表 6.17. swift

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
connect-retries = None	整数值	应该尝试连接错误的最大重试次数。
connect-retry-delay = None	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
container = ironic-inspector	字符串值	创建对象时要使用的默认 Swift 容器。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
delete_after = 0	整数值	Swift 对象在删除前持续的秒数。（设置为 0，永远不会删除对象）。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名

配置选项 = 默认值	类型	Description
endpoint-override = None	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version , min-version , 和/或 max-version 选项。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
max-version = None	字符串值	给定 API 的最大主要版本，旨在作为带有 min_version 的范围的上限。与版本相互排斥。
min-version = None	字符串值	给定 API 的最低主要版本，旨在作为 max_version 的范围低限。与版本相互排斥。如果给出了 min_version，且没有 max_version，它就如同 max version 为 "latest"。
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = object-store	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
system-scope = None	字符串值	系统操作的范围

配置选项 = 默认值	类型	Description
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。
version = None	字符串值	给定主版本 API 版本内用于端点 URL 发现的最低 Major API 版本。与 min_version 和 max_version 相互排斥

第 7 章 KEYSTONE

下面的章节包含有关 **keystone** 服务中的配置选项的信息。

7.1. KEYSTONE.CONF

本节包含 `/etc/keystone/keystone.conf` 文件的选项。

7.1.1. DEFAULT

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
admin_token = None	字符串值	不建议使用 此功能 。反之，使用 keystone-manage bootstrap 命令。此选项的值被视为"共享机密"，可用于通过 API 引导 Keystone。此"令牌"并不代表一个用户（没有身份），也无显式授权（有效地绕过大多数授权检查）。如果设置为 None ，则忽略该值，并且 admin_token 中间件被有效地禁用。
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
control_exchange = keystone	字符串值	默认交换，哪些主题在哪些方面有范围。可以被 <code>transport_url</code> 选项指定的交换名称覆盖。
debug = False	布尔值	如果设置为 <code>true</code> ，则日志级别将设置为 <code>DEBUG</code> ，而不是默认的 <code>INFO</code> 级别。

配置选项 = 默认值	类型	Description
<pre>default_log_levels = ['amqp=WARN', 'amqplib=WARN', 'boto=WARN', 'qpids=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']</pre>	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 log_config_append，则忽略此选项。
default_publisher_id = None	字符串值	用于传出通知的默认 publisher_id。如果未定义，则 Keystone 将默认使用服务器的主机名。
executor_thread_pool_size = 64	整数值	当 executor 为 threading 或 eventlet 时，executor 线程池的大小。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
insecure_debug = False	布尔值	如果设置为 true，则服务器将在 HTTP 响应中返回信息，该响应可能允许未经身份验证的或经过身份验证的用户获得比 normal 更多的信息，如有关身份验证失败的原因的更多详情。这对调试可能很有用，但不安全。
`instance_format = [instance: %(uuid)s]`	字符串值	通过日志消息传递的实例格式。
`instance_uuid_format = [instance: %(uuid)s]`	字符串值	通过日志消息传递的实例 UUID 的格式。

配置选项 = 默认值	类型	Description
list_limit = None	整数值	集合中返回的最大实体数。然后，可以通过在适当的部分中指定 <code>list_limit</code> 来覆盖这个全局限制（例如 [assignment] ）。默认情况下不设置任何限制。在大型部署中，建议您将其设置为合理的数字，以防止列出所有用户和项目等操作，从而给系统造成不必要的负载。
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 <code>log-date-format</code> ）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime)s.%(msecs)03d.%(process)d.%(levelname)s.%(name)s.%(request_id)s.%(user_identity)s.%(instance)s.%(message)s	字符串值	使用上下文格式化用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_debug_format_s uffix = %(funcName)s.%(pathname)s.%(lineno)d	字符串值	当消息的日志记录级别为 <code>DEBUG</code> 时，要附加到日志消息的其他数据。 oslo_log.formatters.ContextFormatter 使用

配置选项 = 默认值	类型	Description
logging_default_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_format = %(user) s %(tenant) s %(domain) s % (user_domain) s % (project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
max_param_size = 64	整数值	限制用户和项目 ID/名称的大小。
max_project_tree_depth = 5	整数值	项目层次结构的最大深度，不包括作为层次结构顶部的域的项目。警告：将其设置为大的值可能会对性能造成负面影响。
max_token_size = 255	整数值	与 [DEFAULT] max_param_size 类似，但为令牌值提供一个例外。使用 Fernet 令牌时，这可设置为低 255。
notification_format = cadf	字符串值	定义身份服务事件的通知格式。 基本 通知仅包含有关正在操作的资源的信息。 cadf 通知具有相同的信息，以及有关事件的启动器的信息。 cadf 选项完全与 基本 选项向后兼容，但完全符合 CADF，建议在审计用例中使用。

配置选项 = 默认值	类型	Description
notification_opt_out = ['identity.authenticate.success', 'identity.authenticate.pending', 'identity.authenticate.failed']	多值	您可以通过显式选择减少通知 keystone 发送的数量。Keystone 不会发出与此列表中表达的模式匹配的通知。值应该采用 identity.<resource_type>.<operation> 的形式。默认情况下，所有与身份验证相关的通知都会被自动隐藏。此字段可以设置多次，以便选择不使用多个通知主题。例如，以下禁止通知描述用户创建或成功身份验证事件： notification_opt_out=identity.user.create notification_opt_out=identity.authenticate.success
public_endpoint = None	URI 值	向客户端公告的 Keystone 的基本公共端点 URL（注意：这不会影响 Keystone 侦听连接的方式）。默认为请求的基域 URL。例如，如果 keystone 收到对 http://server:5000/v3/users 的请求，则此选项将自动视为 http://server:5000 。只有基本 URL 的值包含 keystone 不会自动推断的路径(prefix/v3)或者应该在主机上找到端点时，才应设置选项。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
rpc_conn_pool_size = 30	整数值	RPC 连接池的大小。
rpc_ping_enabled = False	布尔值	添加端点以回答 ping 调用。端点名为 oslo_rpc_server_ping
rpc_response_timeout = 60	整数值	从调用等待响应的秒数。
strict_password_check = False	布尔值	如果设置为 true，则对密码操作执行严格的密码长度检查。如果密码超过最大长度，则操作将失败，并显示 HTTP 403 Forbidden 错误。如果设置为 false，密码会自动截断到最大长度。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。

配置选项 = 默认值	类型	Description
transport_url = rabbit://	字符串值	用于连接到消息传递后端的网络地址和可选用户凭证，采用 URL 格式。预期格式为： driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query 示例：rabbit://rabbitmq:password@127.0.0.1:5672// 有关 URL 中字段的详情，请查看 oslo_messaging.TransportURL 文档 https://docs.openstack.org/oslo.messaging/latest/reference/transport.html
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 log_file 选项并使用 Linux 平台时，才有意义。如果设置了 log_config_append，则忽略此选项。

7.1.2. application_credential

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[application_credential]` 组下可用的选项。

表 7.1. application_credential

配置选项 = 默认值	类型	Description
cache_time = None	整数值	缓存应用程序凭证数据的时间（以秒为单位）。除非启用了全局缓存，否则这无效。

配置选项 = 默认值	类型	Description
 caching = True	布尔值	切换应用程序凭证缓存。除非启用了全局缓存，否则这无效。
 driver = sql	字符串值	keystone.application_credential 命名空间中的应用程序凭证后端驱动程序的入口点。Keystone 仅提供 sql 驱动程序，因此除非您提供了自定义入口点，否则没有更改此设置的原因。
 user_limit = -1	整数值	允许用户创建的最大应用程序凭证数。值为 -1 表示无限。如果没有设置限制，则允许用户创建应用程序凭证，这可能会在 keystone 数据库中导致禁止，或者打开 keystone 攻击。

7.1.3. 分配

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[assignment]** 组下可用的选项。

表 7.2. 分配

配置选项 = 默认值	类型	Description
 driver = sql	字符串值	在 keystone.assignment 命名空间中的分配后端驱动程序（角色分配的位置）的入口点。keystone 本身仅提供 SQL 驱动程序。除非为 keystone 编写专有驱动程序，否则您不需要设置这个选项。
 prohibited_implied_role = ['admin']	列表值	禁止成为代表角色的角色名称列表。

7.1.4. auth

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[auth]** 组下可用的选项。

表 7.3. auth

配置选项 = 默认值	类型	Description
 application_credential = None	字符串值	keystone.auth.application_credential 命名空间中的 <code>application_credential</code> auth 插件模块的入口点。除非您要覆盖 keystone 自己的 application_credential 身份验证插件，否则您不需要设置它。

配置选项 = 默认值	类型	Description
external = None	字符串值	keystone.auth.external 命名空间中的 external (REMOTE_USER) auth 插件模块的入口点。提供的驱动程序是 DefaultDomain 和 Domain 。默认驱动程序为 DefaultDomain ，它假定默认域上下文中存在的 REMOTE_USER 变量中的 keystone 指定的用户名标识的所有用户。 Domain 选项要求将额外的环境变量显示给 keystone, REMOTE_DOMAIN ，其中包含 REMOTE_USER 的域名（如果未设置 REMOTE_DOMAIN ，则改为使用默认域）。除非您要利用“外部身份验证”，否则您不需要设置它，其中应用服务器（如 Apache）正在处理身份验证而不是 keystone。
mapped = None	字符串值	keystone.auth.mapped 命名空间中的映射 auth 插件模块的入口点。除非您要覆盖 keystone 自己的 映射 身份验证插件，否则您不需要设置它。
methods = ['external', 'password', 'token', 'oauth1', 'mapped', 'application_credential']	列表值	允许的身份验证方法。注：如果您当前使用联邦，您应该禁用 外部 验证方法。外部身份验证和联邦都使用 REMOTE_USER 变量。由于调用 mapped 和 external 插件来验证请求环境中的属性，所以可能会导致冲突。
oauth1 = None	字符串值	keystone.auth.oauth1 命名空间中的 OAuth 1.0a auth 插件模块的入口点。除非您要覆盖 keystone 自己的 oauth1 身份验证插件，否则您不需要设置它。
password = None	字符串值	keystone.auth.password 命名空间中的 password auth 插件模块的入口点。除非您要覆盖 keystone 自己的密码身份验证插件，否则您不需要设置它。
token = None	字符串值	keystone.auth.token 命名空间中令牌 auth 插件模块的入口点。除非您要覆盖 keystone 自己的令牌身份验证插件，否则您不需要设置它。

7.1.5. 缓存

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[cache]` 组下可用的选项。

表 7.4. 缓存

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
backend = dogpile.cache.null	字符串值	缓存后端模块。对于具有数百个线程服务器的基于 eventlet 的环境，建议使用带有池 (oslo_cache.memcache_pool) 的 Memcache。对于小于 100 个线程服务器的环境，建议使用 Memcached (dogpile.cache.memcached) 或 Redis (dogpile.cache.redis)。具有单一服务器实例的测试环境可以使用 dogpile.cache.memory 后端。
backend_argument = []	多值	为 backend 模块提供的参数。指定每参数传递到 dogpile.cache 后端的一次这个选项。示例格式："<argname>:<value>"。
config_prefix = cache.oslo	字符串值	为缓存区域构建配置字典的前缀。除非存在另一个具有相同配置名称的 dogpile.cache 区域，否则这不应更改。
dead_timeout = 60	浮点值	尝试在 HashClient 内部机制中重新添加节点前的时间（以秒为单位）。
debug_cache_backend = False	布尔值	从缓存后端进行额外的调试（缓存键、get/set/delete/etc 调用）。只有在您需要使用键/值查看特定的 cache-backend get/set/delete 调用时才有用。通常，这应该设为 false。
enable_retry_client = False	布尔值	启用重试客户端机制来处理失败。这些机制可用于嵌套所有的 pymemcache 客户端。打包程序允许您定义有多少尝试进行，以及 attempts 之间等待的时间。
enable_socket_keepalive = False	布尔值	dogpile 的 pymemcache 后端的套接字 keepalive 的全局切换
enabled = True	布尔值	缓存的全局切换。
expiration_time = 600	整数值	dogpile.cache 区域中的任何缓存项的默认 TTL（以秒为单位）。这适用于任何未为其定义显式缓存过期时间的缓存方法。
hashclient_retry_attempts = 2	整数值	在客户端标记为死前应尝试的次数，并在 HashClient 的内部机制中从池中删除。
hashclient_retry_delay = 1	浮点值	在 HashClient 内部机制中重试尝试之间应经过的时间（以秒为单位）。
memcache_dead_retry = 300	整数值	在再次尝试前，memcached 服务器被视为 dead 的秒数。(dogpile.cache.memcache 和 oslo_cache.memcache_pool 后端)。

配置选项 = 默认值	类型	Description
memcache_pool_connection_get_timeout = 10	整数值	操作将等待的秒数，以获取 memcache 客户端连接。
memcache_pool_flush_on_reconnect = False	布尔值	如果 memcache 在重新连接时清除，则全局切换。（仅oslo_cache.memcache_pool 后端）。
memcache_pool_maxsize = 10	整数值	每个 memcached 服务器的最大打开连接数。（oslo_cache.memcache_pool backend）。
memcache_pool_unused_timeout = 60	整数值	到 memcached 的连接在池关闭前没有被使用的秒数。（仅oslo_cache.memcache_pool 后端）。
memcache_servers = ['localhost:11211']	列表值	memcache 服务器格式为 "host:port"。（dogpile.cache.memcached 和 oslo_cache.memcache_pool 后端）。如果给定主机引用 IPv6 或给定域，则您应该为给定地址系列(inet6) (inet6) (inet6) (inet6[::1]:11211,inet6:[fd12:3456:789a:1::1]:11211,inet6:[controller-0.internalapi]:11211)作为前缀。如果未提供地址系列，则使用的默认地址系列将是 inet （对应于 IPv4）
memcache_socket_timeout = 1.0	浮点值	每次对服务器调用的超时时间（以秒为单位）。（dogpile.cache.memcache 和 oslo_cache.memcache_pool 后端）。
proxies = []	列表值	用于导入的代理类会影响 dogpile.cache 后端功能的方式。有关 change-backend-behavior 的 dogpile.cache 文档。
retry_attempts = 2	整数值	在失败前尝试操作的次数。
retry_delay = 0	浮点值	每次尝试之间休眠的秒数。
socket_keepalive_count = 1	整数值	在丢弃连接前，保持 TCP 的最大 keepalive 探测 TCP 数量。应该是大于零的正整数。
socket_keepalive_idle = 1	整数值	在 TCP 开始发送 keepalive 探测前，连接需要保持闲置的时间（以秒为单位）。应该是大于零的正整数。
socket_keepalive_interval = 1	整数值	各个 keepalive 探测之间的时间（以秒为单位）。应该是大于零的正整数。
tls_allowed_ciphers = None	字符串值	为使用 TLS 上下文创建的套接字设置可用密码。它应该是 OpenSSL 密码列表格式的字符串。如果没有指定，则所有启用了 OpenSSL 的密码都可用。

配置选项 = 默认值	类型	Description
tls_cafile = None	字符串值	以 PEM 格式串联的 CA 证书文件的路径，以建立缓存服务器的真实性。如果 <code>tls_enabled</code> 是 <code>False</code> ，这个选项将被忽略。
tls_certfile = None	字符串值	到 PEM 格式的单个文件的路径，其中包含客户端证书以及建立证书真实性所需的任意数量的 CA 证书。只有在需要客户端身份验证时才需要此文件。如果 <code>tls_enabled</code> 是 <code>False</code> ，这个选项将被忽略。
tls_enabled = False	布尔值	与缓存服务器组合时，TLS 使用的全局切换。
tls_keyfile = None	字符串值	到包含客户端私钥的单个文件的路径。其他私钥将从 <code>tls_certfile</code> 中指定的文件获取。如果 <code>tls_enabled</code> 是 <code>False</code> ，这个选项将被忽略。

7.1.6. catalog

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[catalog]` 组下可用的选项。

表 7.5. catalog

配置选项 = 默认值	类型	Description
cache_time = None	整数值	缓存目录数据的时间（以秒为单位）。除非同时启用了全局和目录缓存，否则这不起作用。目录数据（服务、端点等）通常不会频繁更改，因此可能需要更长的时间超过全局默认值。
caching = True	布尔值	切换目录缓存。除非启用了全局缓存，否则这无效。在典型的部署中，没有理由禁用此功能。
driver = sql	字符串值	keystone.catalog 命名空间中的目录驱动程序的入口点。Keystone 提供了一个 sql 选项（通过 SQL 支持基本 CRUD 操作）、 模板 选项（从磁盘上的模板目录文件加载目录）和 endpoint_filter.sql 选项（支持每个项目的任意服务目录）。
list_limit = None	整数值	在目录集合中返回的最大实体数。通常没有设置此设置的原因，因为部署可以有足够的服务或端点超过合理的限制。
template_file = default_catalog.templates	字符串值	用于模板目录后端的文件的绝对路径。只有在 <code>[catalog]</code> 驱动程序 设为 templated 时，才使用此选项。

7.1.7. CORS

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[cors]` 组下可用的选项。

表 7.6. CORS

配置选项 = 默认值	类型	Description
allow_credentials = True	布尔值	指明实际请求可以包含用户凭证
allow_headers = ['X-Auth-Token', 'X-Openstack-Request-Id', 'X-Subject-Token', 'X-Project-Id', 'X-Project-Name', 'X-Project-Domain-Id', 'X-Project-Domain-Name', 'X-Domain-Id', 'X-Domain-Name', 'Openstack-Auth-Receipt']	列表值	指定在实际请求过程中可以使用哪些标头字段名称。
allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH']	列表值	指明在实际请求期间可以使用哪些方法。
allowed_origin = None	列表值	指明此资源是否可以与请求"origin"标头中收到的域共享。格式："<protocol>://<host>[:<port>]", 没有尾部斜杠。示例： https://horizon.example.com
expose_headers = ['X-Auth-Token', 'X-Openstack-Request-Id', 'X-Subject-Token', 'Openstack-Auth-Receipt']	列表值	指明哪些标头可以安全地公开给 API。默认为 HTTP Simple Headers。
max_age = 3600	整数值	CORS preflight 请求的最大缓存期限。

7.1.8. credential

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[credential]` 组下可用的选项。

表 7.7. credential

配置选项 = 默认值	类型	Description
auth_ttl = 15	整数值	签名 EC2 或 S3 令牌请求的时间长度（以分钟为单位）从令牌请求中包含的时间戳有效。
cache_time = None	整数值	缓存凭证数据的时间（以秒为单位）。除非启用了全局缓存，否则这无效。
caching = True	布尔值	只在检索用户凭证时切换缓存。除非启用了全局缓存，否则这无效。

配置选项 = 默认值	类型	Description
driver = sql	字符串值	keystone.credential 命名空间中的凭据后端驱动程序程序的入口点。Keystone 仅提供 sql 驱动程序，因此除非您提供自定义入口点，否则没有更改此设置的原因。
key_repository = /etc/keystone/credential-keys/	字符串值	包含用于加密和解密存储在凭据后端中的凭据的 Fernet 密钥目录。用于加密凭据的 Fernet 密钥与用于加密 Fernet 令牌的 Fernet 密钥没有关系。这两个密钥集都应单独管理，并且需要不同的轮转策略。不要与用于管理 Fernet 令牌密钥的存储库共享此存储库。
provider = fernet	字符串值	keystone.credential.provider 命名空间中的凭证加密和解密操作的入口点。Keystone 仅提供 fernet 驱动程序，因此，除非提供了用于加密和解密凭据的自定义入口点，否则没有更改此设置的原因。
user_limit = -1	整数值	允许用户创建的最大凭证数。值为 -1 表示无限。如果没有设置限制，则允许用户在其上创建凭证，这可能会导致 keystone 数据库中的 bloat 或打开 keystone 攻击。

7.1.9. database

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[database]` 组下可用的选项。

表 7.8. database

配置选项 = 默认值	类型	Description
backend = sqlalchemy	字符串值	用于数据库的后端。
connection = None	字符串值	用于连接到数据库的 SQLAlchemy 连接字符串。
connection_debug = 0	整数值	SQL 调试信息的详细程度：0=None, 100=Everything。
<code>`connection_parameters = `</code>	字符串值	用于连接时附加到连接 URL 的可选 URL 参数；指定 <code>param1=value1&param2=value2&...</code>
connection_recycle_time = 3600	整数值	连接池中存在的连接的时间超过这个秒数，下一次从池中签出它们的时间将替换为新的连接。
connection_trace = False	布尔值	将 Python 堆栈跟踪作为注释字符串添加到 SQL。
db_inc_retry_interval = True	布尔值	如果为 True，将数据库操作的重试间隔增加到 <code>db_max_retry_interval</code> 。

配置选项 = 默认值	类型	Description
db_max_retries = 20	整数值	在引发错误前出现连接错误或死锁错误时的最大重试次数。设置为 -1 以指定无限重试计数。
db_max_retry_interval = 10	整数值	如果设置了 db_inc_retry_interval，则数据库操作重试之间的最大秒数。
db_retry_interval = 1	整数值	数据库事务重试之间的秒数。
max_overflow = 50	整数值	如果设置，请将这个值用于 SQLAlchemy 的 max_overflow。
max_pool_size = 5	整数值	在池中保留的最大 SQL 连接数。将值设为 0 表示没有限制。
max_retries = 10	整数值	启动期间数据库连接重试的最大数量。设置为 -1 以指定无限重试计数。
mysql_enable_ndb = False	布尔值	如果为 True，则透明地启用对处理 MySQL 集群 (NDB) 的支持。
mysql_sql_mode = TRADITIONAL	字符串值	用于 MySQL 会话的 SQL 模式。这个选项（包括默认选项）会覆盖任何 server-set SQL 模式。要使用服务器配置设置的任何 SQL 模式，请将其设置为 no 值。 示例：mysql_sql_mode=
pool_timeout = None	整数值	如果设置，请将这个值用于 SQLAlchemy 的 pool_timeout。
retry_interval = 10	整数值	重新尝试打开 SQL 连接之间的间隔。
slave_connection = None	字符串值	用于连接到从数据库的 SQLAlchemy 连接字符串。
sqlite_synchronous = True	布尔值	如果为 True，则 SQLite 使用同步模式。
use_db_reconnect = False	布尔值	启用在连接丢失时进行数据库重新连接的实验性使用。

7.1.10. domain_config

下表概述了 /etc/keystone/keystone.conf 文件中的 [domain_config] 组下可用的选项。

表 7.9. domain_config

配置选项 = 默认值	类型	Description
cache_time = 300	整数值	用于缓存域特定配置数据的生存时间(TTL, 以秒为单位)。除非启用了 [domain_config] 缓存, 否则这不起作用。
caching = True	布尔值	切换到域特定配置后端的缓存。除非启用了全局缓存, 否则这无效。通常没有禁用此功能的原因。
driver = sql	字符串值	keystone.resource.domain_config 命名空间中的特定于域的配置驱动程序的入口点。keystone 只提供一个 sql 选项, 因此除非您提供了自定义入口点, 否则没有理由设置它。

7.1.11. endpoint_filter

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[endpoint_filter]** 组下可用的选项。

表 7.10. endpoint_filter

配置选项 = 默认值	类型	Description
driver = sql	字符串值	keystone.endpoint_filter 命名空间中的端点过滤器驱动程序的入口点。keystone 只提供一个 sql 选项, 因此除非您提供了自定义入口点, 否则没有理由设置它。
return_all_endpoints_if_no_filter = True	布尔值	如果配置的端点过滤器不生成用户 + 项目对的任何端点 (因此可能会为空的服务目录), 这会控制 keystone 的行为。如果设置为 true, 则 keystone 将返回整个服务目录。如果设置为 false, 则 keystone 将返回一个空服务目录。

7.1.12. endpoint_policy

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[endpoint_policy]** 组下可用的选项。

表 7.11. endpoint_policy

配置选项 = 默认值	类型	Description
driver = sql	字符串值	keystone.endpoint_policy 命名空间中的端点策略驱动程序的入口点。keystone 只提供一个 sql 驱动程序, 因此除非您提供自定义入口点, 否则没有设置它的原因。

7.1.13. eventlet_server

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[eventlet_server]** 组下可用的选项。

表 7.12. eventlet_server

配置选项 = 默认值	类型	Description
admin_bind_host = 0.0.0.0	主机地址值	要侦听的 admin 服务的网络接口的 IP 地址。弃用自：K *reason:*Support 在 eventlet 下运行 keystone 已在 Newton 发行版本中删除。这些选项保持向后兼容，因为它们用于 URL 替换。
admin_port = 35357	端口值	要侦听的 admin 服务的端口号。弃用自：K *reason:*Support 在 eventlet 下运行 keystone 已在 Newton 发行版本中删除。这些选项保持向后兼容，因为它们用于 URL 替换。
public_bind_host = 0.0.0.0	主机地址值	要侦听的公共服务的网络接口的 IP 地址。弃用自：K *reason:*Support 在 eventlet 下运行 keystone 已在 Newton 发行版本中删除。这些选项保持向后兼容，因为它们用于 URL 替换。
public_port = 5000	端口值	要侦听的公共服务的端口号。弃用自：K *reason:*Support 在 eventlet 下运行 keystone 已在 Newton 发行版本中删除。这些选项保持向后兼容，因为它们用于 URL 替换。

7.1.14. federation

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[federation]` 组下可用的选项。

表 7.13. federation

配置选项 = 默认值	类型	Description
<code>`assertion_prefix = `</code>	字符串值	过滤环境变量名称进行联邦断言时使用的前缀。匹配的变量被传递给联邦映射引擎。
caching = True	布尔值	切换到联邦缓存。除非启用了全局缓存，否则这无效。通常没有禁用此功能的原因。
default_authorization_ttl = 0	整数值	从映射传输的组成员资格的有效性（以分钟为单位）的默认时间（以分钟为单位）。默认值为 0，这意味着禁用。
driver = sql	字符串值	keystone.federation 命名空间中的联邦后端驱动程序程序的入口点。Keystone 仅提供 sql 驱动程序，因此不要设置此选项，除非您提供了自定义入口点。

配置选项 = 默认值	类型	Description
federated_domain_name = Federated	字符串值	保留允许联邦临时用户具有域概念的任意域名。请注意，管理员将无法使用此名称创建域，或将现有域更新为此名称。除非确实需要修改这个值，否则不建议更改这个值。 弃用自：T <i>*reason:*</i> 这个选项已被其身份提供程序域中存在的临时用户替代。
remote_id_attribute = None	字符串值	用于从环境中获取身份提供程序的实体 ID 的所有协议的默认值。对于 mod_shib ，这将是 Shib-Identity-Provider 。对于 mod_auth_openidc ，这可能是 HTTP_OIDC_ISS 。对于 mod_auth_mellon ，这可能是 MELLON_IDP 。这可以通过使用 API 向联邦协议提供 remote_id_attribute 来基于每个协议覆盖。
sso_callback_template = /etc/keystone/sso_callback_template.html	字符串值	用作单点登录回调处理程序的 HTML 文件的绝对路径。本页预期通过将令牌编码到 POST 请求，将来自 keystone 的用户重定向到可信仪表板主机。Keystone 的默认值应该足以满足大多数部署的需要。
trusted_dashboard = []	多值	可信仪表板主机列表。在接受单点登录请求以返回令牌之前，原始主机必须是此列表的成员。对于多个值，此配置选项可以重复。您必须将其设置为使用基于 Web 的 SSO 流。例如： trusted_dashboard=https://acme.example.com/auth/webssso trusted_dashboard=https://beta.example.com/auth/webssso

7.1.15. fernet_receipts

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[fernet_receipts]` 组下可用的选项。

表 7.14. fernet_receipts

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
key_repository = <code>/etc/keystone/fernet-keys/</code>	字符串值	包含 Fernet receipt 密钥的目录。在第一次使用 keystone-manage fernet_setup 前这个目录需要已存在，则对于运行 keystone-manage fernet_setup 或 keystone-manage fernet_rotate 的用户是可写的，则对于 keystone 的服务器进程是可读的。存储库可以在三个状态之一中包含键：单个暂存密钥（始终索引 0）用于接收验证、一个用于接收和验证的主键（始终为最高索引），以及用于接收验证的辅助键（所有其他索引值）的数量。对于多个 keystone 节点，每个节点必须共享相同的密钥存储库内容，但暂存密钥除外(index 0)。在任意一个节点上运行 keystone-manage fernet_rotate 后可以安全地将暂存密钥(index 0)提升为新主要（从上一个最高索引开始），生成一个新的暂存密钥（带有 index 0 的新密钥）；生成的存储库可以原子复制到其他节点，而无需出现竞争条件的风险（例如：在主机 A 上运行 keystone-manage fernet_rotate 是安全的，等待任何时间，在主机 A 上创建一个 tarball，在主机 B 上解包它到临时位置，并将目录移动(mv)到主机 B 上的目录。在密钥存储库中运行 keystone-manage fernet_rotate 两次而不同步其他节点，则会导致无法被所有节点验证。
max_active_keys = 3	整数值	这会控制 keystone-manage fernet_rotate 轮转中保留多少密钥，然后再丢弃它们。默认值 3 表示 keystone 将维护一个暂存密钥（始终索引 0）、一个主密钥（最高数字索引）和一个辅助键（每个索引）。增加这个值意味着将保存在轮转中。

7.1.16. fernet_tokens

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[fernet_tokens]` 组下可用的选项。

表 7.15. fernet_tokens

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
key_repository = <code>/etc/keystone/fernet-keys/</code>	字符串值	包含 Fernet 令牌密钥的目录。在第一次使用 keystone-manage fernet_setup 前这个目录需要已存在，则对于运行 keystone-manage fernet_setup 或 keystone-manage fernet_rotate 的用户是可写的，则对于 keystone 的服务器进程是可读的。存储库可以在三种状态之一中包含键：单个暂存密钥（始终索引 0）用于令牌验证、用于令牌验证的单一主密钥（始终为最高索引），以及用于令牌验证的辅助键（所有其他索引值）。对于多个 keystone 节点，每个节点必须共享相同的密钥存储库内容，但暂存密钥除外(index 0)。在任意一个节点上运行 keystone-manage fernet_rotate 后可以安全地将暂存密钥(index 0)提升为新主要（从上一个最高索引开始），生成一个新的暂存密钥（带有 index 0 的新密钥）；生成的存储库可以原子复制到其他节点，而无需出现竞争条件的风险（例如：在主机 A 上运行 keystone-manage fernet_rotate 是安全的，等待任何时间，在主机 A 上创建一个 tarball，在主机 B 上解包它到临时位置，并将目录移动(mv)到主机 B 上的目录。在密钥存储库上运行 keystone-manage fernet_rotate 两次，而不同步其他节点，则会导致令牌无法被所有节点验证。
max_active_keys = 3	整数值	这会控制 keystone-manage fernet_rotate 轮转中保留多少密钥，然后再丢弃它们。默认值 3 表示 keystone 将维护一个暂存密钥（始终索引 0）、一个主密钥（最高数字索引）和一个辅助键（每个索引）。增加这个值意味着将保存在轮转中。

7.1.17. healthcheck

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[healthcheck]** 组下可用的选项。

表 7.16. healthcheck

配置选项 = 默认值	类型	Description
backends = []	列表值	可执行健康检查并报告该信息作为请求的一部分的其他后端。
detailed = False	布尔值	显示响应中更详细的信息。安全备注：启用此选项可能会公开有关被监控服务的敏感详情。务必验证它是否不会违反您的安全策略。
disable_by_file_path = None	字符串值	检查是否存在文件，以确定应用是否在端口上运行。DisableByFileHealthcheck 插件使用。

配置选项 = 默认值	类型	Description
<code>disable_by_file_paths = []</code>	列表值	根据端口检查文件是否存在，以确定应用是否在端口上运行。需要字符串的 "port:path" 列表。 DisableByFilesPortsHealthcheck 插件使用。
<code>path = /healthcheck</code>	字符串值	响应 healthcheck 请求的路径。

7.1.18. identity

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[identity]` 组下可用的选项。

表 7.17. identity

配置选项 = 默认值	类型	Description
<code>cache_time = 600</code>	整数值	缓存身份数据的时间（以秒为单位）。除非启用了全局和身份缓存，否则这不起作用。
<code>caching = True</code>	布尔值	切换身份缓存。除非启用了全局缓存，否则这无效。通常没有禁用此功能的原因。
<code>default_domain_id = default</code>	字符串值	这引用了用于所有身份 API v2 请求的域（不知道域）。 keystone-manage bootstrap 可以选择性地为您创建具有此 ID 的域。此 ID 引用的域无法在 v3 API 上删除，以防止意外破坏 v2 API。对此域没有特别的，除了必须存在的，才能保持对 v2 客户端的支持。通常没有更改此值的原因。
<code>domain_config_dir = /etc/keystone/domains</code>	字符串值	keystone 应定位域特定 <code>[identity]</code> 配置文件的绝对路径。除非将 <code>[identity] domain_specific_drivers_enabled</code> 设置为 true，否则此选项无效。通常没有更改此值的原因。
<code>domain_configurations_from_database = False</code>	布尔值	默认情况下，特定于域的配置数据是从 <code>[identity] domain_config_dir</code> 标识的目录中的文件读取。启用此配置选项允许您通过 API 管理域特定配置，这些配置在后端（通常是 SQL 数据库）中保留，而不是使用磁盘上的配置文件。
<code>domain_specific_drivers_enabled = False</code>	布尔值	域的子集（或全部）可以有自己的身份驱动程序，每个驱动程序都有自己的部分配置选项，存储在资源后端或域配置目录中的文件中（取决于 <code>[identity] domain_configurations_from_database</code> 的设置）。只有特定于域的值才需要以这种方式指定。此功能默认为禁用，但可能在以后的发行版本中默认启用，设置为 true 以启用。

配置选项 = 默认值	类型	Description
driver = sql	字符串值	keystone.identity 命名空间中的身份后端驱动程序的入口点。Keystone 提供 sql 和 ldap 驱动程序。如果启用了 [identity] domain_specific_drivers_enabled , 则此选项也用作默认驱动程序选择 (以及本节中的其他配置)。除非您的部署主要依赖于 ldap AND 不使用域特定配置, 否则通常应将此设置为 sql 。
list_limit = None	整数值	在身份集合中返回的最大实体数。
max_password_length = 4096	整数值	用户密码允许的最大长度。降低这个值以提高性能。更改此值不会影响现有密码。这个值也可以被特定的哈希算法最大允许长度覆盖, 其优先于配置的值。bcrypt max_password_length 为 72 字节。
password_hash_algorithm = bcrypt	字符串值	用于存储在 keystone 中的密码的密码哈希算法。
password_hash_rounds = None	整数值	这个选项代表了在安全性和性能之间权衡。数值越高, 性能会较慢, 但安全性更高。更改此选项将只影响新创建的密码, 因为已应用固定的轮换数, 因此可以安全地在正在运行的集群中调整这个选项。bcrypt 的默认值为 12, 必须在 4 到 31 之间, 包括: scrypt 的默认值为 16, 必须在 range (1,32) 内。pbkdf_sha512 的默认值为 60000, 必须在 范围 (1,1<<32) WARNING 中: 如果使用 scrypt, 增加这个值会增加 BOTH 时间和内存要求来哈希密码。
salt_bytesize = None	整数值	在 scrypt 和 pbkdf2_sha512 哈希 salt 中使用的字节数。scrypt 的默认值为 16 字节。pbkdf2_sha512 默认为 16 字节。限制为最多 96 字节, 因为用于存储密码哈希的列的大小。
scrypt_block_size = None	整数值	要传递给 scrypt 哈希功能的可选块大小(r 参数)。对于调整 scrypt 以优化 CPU 架构性能非常有用。只有在 password_hash_algorithm 选项设置为 scrypt 时, 才会使用此选项。默认值为 8。
scrypt_parallelism = None	整数值	可选的 parallelism 传递给 scrypt 哈希函数(p 参数)。只有在 password_hash_algorithm 选项设置为 scrypt 时, 才会使用此选项。默认为 1。

7.1.19. identity_mapping

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[identity_mapping]** 组下可用的选项。

表 7.18. identity_mapping

配置选项 = 默认值	类型	Description
backward_compatible_ids = True	布尔值	对于没有生成 UUID（如 LDAP）的后端，用户和组群 ID 的格式在 Juno 中被改变，它带有提供到 LDAP 中底层属性的哈希映射的 keystone。默认情况下禁用此映射，这样可确保现有 ID 不会改变。即使映射通过使用域特定驱动程序([identity] domain_specific_drivers_enabled)启用，但 LDAP 处理的默认域的用户和组仍会不会被映射，以确保其 ID 保持向后兼容性。将此值设置为 false 将为所有后端启用新映射，包括默认的 LDAP 驱动程序。如果您尚未从默认 LDAP 域分配用户和组，且被 Keystone 为客户端提供不同的 ID，则只保证启用此选项(API 中的现有 ID 将会更改)。通常，这意味着您唯一可以将此值设置为 false 的时间是配置全新安装时，但这是推荐的值。
driver = sql	字符串值	keystone.identity.id_mapping 命名空间中的身份映射后端驱动程序的入口点。Keystone 仅提供 sql 驱动程序，因此除非您提供了自定义入口点，否则没有更改此设置的原因。
generator = sha256	字符串值	keystone.identity.id_generator 命名空间中用户和组实体的公共 ID 生成器的入口点。Keystone 身份映射程序仅支持生成 64 字节或更少生成器。Keystone 仅提供 sha256 入口点，因此除非您提供自定义入口点，否则没有更改此值的原因。

7.1.20. jwt_tokens

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[jwt_tokens]** 组下可用的选项。

表 7.19. jwt_tokens

配置选项 = 默认值	类型	Description
jws_private_key_repository = /etc/keystone/jws-keys/private	字符串值	包含用于签名 JWS 令牌的私钥的目录。此目录必须存在，以便 keystone 的服务器进程启动。它还必须可以被 keystone 的服务器进程读取。它必须至少包含一个与 keystone.conf [jwt_tokens] jws_public_key_repository 中对应公钥的私钥。如果此目录中存在多个私钥，则 keystone 将使用名为 private.pem 的密钥来签署令牌。未来，keystone 可以支持使用多个私钥为令牌签名。现在，在这个目录中仅需要名为 private.pem 的密钥才能发布 JWS 令牌。这个选项仅适用于发出 JWS 令牌并设置 keystone.conf [token] provider = jws 。

配置选项 = 默认值	类型	Description
jws_public_key_repositor y = /etc/keystone/jws- keys/public	字符串值	包含用于验证 JWS 令牌签名的公钥的目录。此目录必须存在，以便 keystone 的服务器进程启动。它还必须可以被 keystone 的服务器进程读取。它必须至少包含一个与 keystone.conf [jwt_tokens] jws_private_key_repository 中的私钥对应的公钥。这个选项仅适用于发出 JWS 令牌并设置 keystone.conf [token] provider = jws 。

7.1.21. Ildap

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[ldap]** 组下可用的选项。

表 7.20. Ildap

配置选项 = 默认值	类型	Description
alias_dereferencing = default	字符串值	用于涉及别名的查询的 LDAP 解引用选项。值 default 会返回使用由 ldap.conf 配置的默认取消引用行为。值 never 阻止别名被全部解引用。值 搜索 仅在名称解析后解引用别名。用于在名称解析过程中 查找 别名的值。值 always 在所有情况下都会取消引用别名。
auth_pool_connection_lif etime = 60	整数值	与 LDAP 服务器的最大最终用户身份验证连接生命周期（以秒为单位）。超过这个生命周期时，连接将绑定并从连接池中删除。除非 [ldap] use_auth_pool 也启用，否则此选项无效。
auth_pool_size = 100	整数值	用于最终用户身份验证的连接池的大小。除非 [ldap] use_auth_pool 也启用，否则此选项无效。
chase_referrals = None	布尔值	设置 keystone 在目录分区的引用行为。如果保留未设置，则使用系统的默认行为。
connection_timeout = -1	整数值	与 LDAP 服务器一起使用的连接超时。值 -1 表示连接永远不会超时。
debug_level = None	整数值	为 LDAP 调用设置 LDAP 调试级别。值 0 表示没有启用调试。这个值是一个位掩码，请咨询您的 LDAP 文档了解可能的值。
group_ad_nesting = False	布尔值	如果启用，组查询将使用特定于 Active Directory 的过滤器进行嵌套组。

配置选项 = 默认值	类型	Description
group_additional_attribute_mapping = []	列表值	用于将额外属性映射到 keystone 中的组的属性对的 LDAP 属性列表。预期的格式为 <ldap_attr>: <group_attr> ，其中 ldap_attr 是 LDAP 对象和 group_attr 中的属性，它应当出现在身份 API 中。
group_attribute_ignore = []	列表值	要忽略的 create 和 update 的组属性列表。或者，或者是否应该为列表或显示组过滤特定的组属性。
group_desc_attribute = description	字符串值	在 keystone 中映射到组描述的 LDAP 属性。
group_filter = None	字符串值	用于组的 LDAP 搜索过滤器。
group_id_attribute = cn	字符串值	在 keystone 中映射到组 ID 的 LDAP 属性。这不能是一个多值属性。组 ID 应该在 keystone 域和 URL-safe 之间全局唯一。
group_member_attribute = member	字符串值	用于指示用户是组的成员的 LDAP 属性。
group_members_are_ids = False	布尔值	如果组对象类的成员是 keystone 用户 ID 而不是 LDAP DN，则启用此选项。当使用 posixGroup 作为 Open Directory 中的组对象类时，会出现这种情况。
group_name_attribute = ou	字符串值	在 keystone 中映射到组名称的 LDAP 属性。组名称预期只在 keystone 域中是唯一的，且不应该是 URL-safe。
group_objectclass = groupOfNames	字符串值	用于组的 LDAP 对象类。如果将此选项设置为 posixGroup ，您可能还对启用 [ldap] group_members_are_ids 选项感兴趣。
group_tree_dn = None	字符串值	用于组的搜索基础。默认为 ou=UserGroups ，并附加了 [ldap] 后缀。
page_size = 0	整数值	定义 keystone 在列出对象时从 LDAP 服务器请求的每个页面的最大结果数。值为零(0)禁用分页。
password = None	字符串值	如果您的 LDAP 服务器需要，管理员的密码将绑定 DN 在查询 LDAP 服务器时使用。
pool_connection_lifetime = 600	整数值	到 LDAP 服务器的最大连接生命周期（以秒为单位）。超过这个生命周期时，连接将绑定并从连接池中删除。除非 [ldap] use_pool 也启用，否则此选项无效。

配置选项 = 默认值	类型	Description
pool_connection_timeout = -1	整数值	池 LDAP 连接时使用的连接超时。值 -1 表示连接永远不会超时。除非 [ldap] use_pool 也启用，否则此选项无效。
pool_retry_delay = 0.1	浮点值	尝试重新连接到 LDAP 服务器前等待的秒数。除非 [ldap] use_pool 也启用，否则此选项无效。
pool_retry_max = 3	整数值	在中止前尝试重新连接到 LDAP 服务器的次数上限。值为零可防止重试。除非 [ldap] use_pool 也启用，否则此选项无效。
pool_size = 10	整数值	LDAP 连接池的大小。除非 [ldap] use_pool 也启用，否则此选项无效。
query_scope = one	字符串值	搜索范围，用于定义在搜索基础中如何深度搜索。一个值 one （代表 oneLevel 或 singleLevel ）表示对象立即搜索到基本对象，但不包括基本对象本身。 sub （代表子树或整个 Subtree ）的值表示基础对象本身和其下整个子树的搜索。
randomize_urls = False	布尔值	随机化每个 keystone 进程中的 URL 顺序。这使得失败行为更逐步化，因为如果第一个服务器停机，进程/线程将在尝试连接服务器进一步关闭前等待指定的超时。默认值为 False ，以便向后兼容。
suffix = cn=example,cn=com	字符串值	如果没有通过 [ldap] user_tree_dn 或 [ldap] group_tree_dn 定义 DN，要使用的默认 LDAP 服务器后缀。
tls_cacertdir = None	字符串值	与 LDAP 服务器通信时要使用的 CA 证书目录的绝对路径。如果您也设置了 [ldap] tls_cacertfile ，则没有设置此选项的原因。
tls_cacertfile = None	字符串值	与 LDAP 服务器通信时使用的 CA 证书文件的绝对路径。这个选项优先于 [ldap] tls_cacertdir ，因此不能同时设置这两个选项。
tls_req_cert = demand	字符串值	指定针对传入 TLS 会话上的客户端证书执行的检查。如果设置为 demand ，则 LDAP 服务器将始终请求和要求证书。如果设置为 allow ，则始终向 LDAP 服务器请求证书，但不需要证书。如果设置为 never ，则不会请求证书。
url = ldap://localhost	字符串值	用于连接到 LDAP 服务器的 URL。多个 LDAP URL 可以指定为用逗号分开的字符串。成功绑定的第一个 URL 用于连接。

配置选项 = 默认值	类型	Description
use_auth_pool = True	布尔值	为最终用户身份验证启用 LDAP 连接池。通常没有禁用此功能的原因。
use_pool = True	布尔值	启用 LDAP 连接池以查询 LDAP 服务器。通常没有禁用此功能的原因。
use_tls = False	布尔值	在与 LDAP 服务器通信时启用 TLS。使用此选项时，您还应设置 [ldap] tls_cacertfile 和 [ldap] tls_cacertdir 选项。如果您通过 SSL (LDAPS) 而不是 TLS 使用 LDAP，则不要设置此选项。
user = None	字符串值	如果您的 LDAP 服务器需要，管理员的用户名会绑定 DN 在查询 LDAP 服务器时使用。
user_additional_attribute_mapping = []	列表值	用于向 keystone 用户属性对的 LDAP 属性列表，用于将额外属性映射到 keystone 中的用户。预期的格式为 <ldap_attr>:<user_attr> ，其中 ldap_attr 是 LDAP 对象和 user_attr 中的属性，它应当出现在身份 API 中。
user_attribute_ignore = ['default_project_id']	列表值	要在创建和更新时要忽略的用户属性列表，或者是否为 list 或 show user 过滤特定的用户属性。
user_default_project_id_attribute = None	字符串值	在 keystone 中映射到用户的 default_project_id 的 LDAP 属性。当 keystone 对 LDAP 具有写入访问权限时，这最常使用。
user_description_attribute = description	字符串值	在 keystone 中映射到用户描述的 LDAP 属性。
user_enabled_attribute = enabled	字符串值	在 keystone 中映射到用户启用属性的 LDAP 属性。如果将此选项设置为 userAccountControl ，那么您可能还对设置 [ldap] user_enabled_mask 和 [ldap] user_enabled_default 感兴趣。
user_enabled_default = True	字符串值	启用用户的默认值。如果 LDAP 服务器使用非布尔值 (bitmask) 值来指示用户是否启用或禁用了，则这应匹配适当的整数值。如果这未设置为 True ，则典型的值为 512 。这通常在 [ldap] user_enabled_attribute = userAccountControl 时使用。
user_enabled_emulation = False	布尔值	如果启用，则 keystone 使用一种替代方法来确定用户是否已启用，还是通过检查它们是否是 [ldap] user_enabled_emulation_dn 选项定义的组的成员。启用这个选项会导致 keystone 忽略 [ldap] user_enabled_invert 的值。

配置选项 = 默认值	类型	Description
user_enabled_emulation_dn = None	字符串值	使用启用模拟时用于存放启用的用户的组条目的 DN。设置此选项不会起作用，除非也启用了 [ldap] user_enabled_emulation 。
user_enabled_emulation_use_group_config = False	布尔值	使用 [ldap] group_member_attribute 和 [ldap] group_objectclass 设置来确定模拟启用的组中的成员资格。启用这个选项不会起作用，除非也启用了 [ldap] user_enabled_emulation 。
user_enabled_invert = False	布尔值	在逻辑上对从 LDAP 服务器获取的 enabled 属性的布尔值进行求值。有些 LDAP 服务器使用布尔值锁定属性，其中 "true" 表示帐户被禁用。设置 [ldap] user_enabled_invert = true 将允许使用这些锁定属性。如果使用 [ldap] user_enabled_mask 或 [ldap] user_enabled_emulation 选项，这个选项将无效。
user_enabled_mask = 0	整数值	位掩码整数，以选择哪个位表示启用的值，如果 LDAP 服务器在整数中以位表示为位，而不是作为离散的布尔值。值 0 表示掩码没有被使用。如果这没有设置为 0 ，则典型的值为 2 。这通常在 [ldap] user_enabled_attribute = userAccountControl 时使用。设置这个选项会导致 keystone 忽略 [ldap] user_enabled_invert 的值。
user_filter = None	字符串值	用于用户的 LDAP 搜索过滤器。
user_id_attribute = cn	字符串值	在 keystone 中映射到用户 ID 的 LDAP 属性。这不能是一个多值属性。用户 ID 应该在 keystone 域和 URL-safe 之间全局唯一。
user_mail_attribute = mail	字符串值	在 keystone 中映射到用户电子邮件的 LDAP 属性。
user_name_attribute = sn	字符串值	在 keystone 中映射到用户名的 LDAP 属性。用户名预期只在 keystone 域中是唯一的，且预期为 URL-safe。
user_objectclass = inetOrgPerson	字符串值	用于用户的 LDAP 对象类。
user_pass_attribute = userPassword	字符串值	在 keystone 中映射到用户密码的 LDAP 属性。
user_tree_dn = None	字符串值	用于用户的搜索基础。默认为 ou=Users ，并附加了 [ldap] 后缀。

7.1.22. memcache

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[memcache]` 组下可用的选项。

表 7.21. memcache

配置选项 = 默认值	类型	Description
<code>dead_retry = 300</code>	整数值	在再次尝试前，memcached 服务器被视为 dead 的秒数。这供 key 值存储系统使用。
<code>pool_connection_get_timeout = 10</code>	整数值	操作将等待的秒数，以获取 memcache 客户端连接。这供 key 值存储系统使用。
<code>pool_maxsize = 10</code>	整数值	每个 memcached 服务器的最大开放连接数。这供 key 值存储系统使用。
<code>pool_unused_timeout = 60</code>	整数值	到 memcached 的连接在池关闭前没有被使用的秒数。这供 key 值存储系统使用。
<code>socket_timeout = 3</code>	整数值	每个对服务器的调用的超时时间（以秒为单位）。这供 key 值存储系统使用。弃用自：T *reason:* 这个选项与 <code>oslo.cache</code> 重复。配置 <code>"keystone.conf [cache] memcache_socket_timeout"</code> 选项，以设置 memcached 的 <code>socket_timeout</code> 。

7.1.23. oauth1

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[oauth1]` 组下可用的选项。

表 7.22. oauth1

配置选项 = 默认值	类型	Description
<code>access_token_duration = 86400</code>	整数值	OAuth 访问令牌在创建后保持有效的秒数。这是消费者必须与服务提供商（通常是 keystone）交互所需的时间。将这个选项设置为零意味着访问令牌将持续存在。
<code>driver = sql</code>	字符串值	<code>keystone.oauth1</code> 命名空间中的 OAuth 后端驱动程序的入口点。通常，除非您提供了自定义入口点，否则没有设置这个选项的原因。
<code>request_token_duration = 28800</code>	整数值	OAuth Request Token 在创建后保持有效的秒数。这是用户必须授权令牌的时间。将这个选项设置为零表示请求令牌将持续存在。

7.1.24. oslo_messaging_amqp

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[oslo_messaging_amqp]` 组下可用的选项。

表 7.23. oslo_messaging_amqp

配置选项 = 默认值	类型	Description
addressing_mode = dynamic	字符串值	表示驱动程序使用的寻址模式。允许的值： <i>legacy</i> - 使用旧的不可路由寻址； <i>routable</i> - 使用可路由地址； <i>dynamic</i> - 如果消息总线不支持路由，使用 <i>legacy</i> ，如果支持，使用 <i>routable</i> 。
anycast_address = anycast	字符串值	在发送到一组消费者时附加到地址前缀。消息总线用来识别应在消费者之间以轮循方式发送的消息。
broadcast_prefix = broadcast	字符串值	广播到所有服务器时使用的地址前缀
connection_retry_backoff = 2	整数值	每次失败故障转移尝试后，将 <code>connection_retry_interval</code> 增加到这秒数。
connection_retry_interval = 1	整数值	尝试重新连接前暂停的秒数。
connection_retry_interval_max = 30	整数值	<code>connection_retry_interval</code> + <code>connection_retry_backoff</code> 的最大限制
container_name = None	字符串值	AMQP 容器的名称必须全局唯一。默认为生成的 UUID
default_notification_exchange = None	字符串值	通知地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_notification_exchange if set else control_exchange if set else <i>notify</i>
default_notify_timeout = 30	整数值	发送通知消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_reply_retry = 0	整数值	重新发送回复消息的最大尝试次数，因为可恢复错误而失败。
default_reply_timeout = 30	整数值	rpc 回复消息发送的截止时间。
default_rpc_exchange = None	字符串值	RPC 地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_rpc_exchange if set other control_exchange if set else <i>rpc</i>
default_send_timeout = 30	整数值	rpc cast 或调用消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_sender_link_timeout = 600	整数值	调度闲置发送者链接清除的持续时间。过期后分离链接。

配置选项 = 默认值	类型	Description
group_request_prefix = unicast	字符串值	发送到组中任何服务器时的地址前缀
idle_timeout = 0	整数值	不活跃连接的超时时间（以秒为单位）
link_retry_delay = 10	整数值	重新连接 AMQP 1.0 链接之间的暂停时间，该链接因为可恢复的错误而失败。
multicast_address = multicast	字符串值	在发送异常消息时，附加到地址前缀。消息总线用来识别 fanout 消息。
notify_address_prefix = openstack.org/om/notify	字符串值	所有生成的通知地址的地址前缀
notify_server_credit = 100	整数值	传入通知消息的窗口大小
pre_settled = ['rpc-cast', 'rpc-reply']	多值	发送此类型的消息预先组装。预先设置的消息不会接收来自 peer 的确认信息。请注意：如果发送失败，则预先设置的消息可能会被静默丢弃。允许的值： <i>rpc-call</i> - 发送 RPC Calls <i>pre-settled</i> <i>rpc-reply</i> - 发送 RPC Replies <i>pre-settled</i> <i>rpc-cast</i> - Send RPC Casts <i>pre-settled</i> <i>notify</i> - Send Notifications <i>pre-settled</i>
pseudo_vhost = True	布尔值	为不支持虚拟主机的消息总线启用虚拟主机支持（如 qpidd）。当设置为 true 时，虚拟主机名将添加到所有消息总线地址中，从而有效地为每个虚拟主机 <i>创建专用子网</i> 。如果消息总线支持使用 AMQP 1.0 Open performative 中的 <i>hostname</i> 字段作为虚拟主机的名称，设置为 False。
reply_link_credit = 200	整数值	传入 RPC 消息的窗口大小。
rpc_address_prefix = openstack.org/om/rpc	字符串值	所有生成的 RPC 地址的地址前缀
rpc_server_credit = 100	整数值	传入 RPC Request 消息的窗口大小
<code>`sas_config_dir = `</code>	字符串值	包含 SASL 配置的目录路径
<code>`sas_config_name = `</code>	字符串值	配置文件名称（不带 .conf 后缀）
<code>`sas_default_realm = `</code>	字符串值	如果用户名中没有 realm，则要使用的 SASL 域
<code>`sas_mechanisms = `</code>	字符串值	可接受的 SASL 机制的以空格分隔的列表

配置选项 = 默认值	类型	Description
server_request_prefix = exclusive	字符串值	发送到特定服务器时使用的地址前缀
ssl = False	布尔值	尝试通过 SSL 连接。如果没有给出其他与 ssl 相关的参数，它将使用系统的 CA-bundle 来验证服务器证书。
<code>`ssl_ca_file = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_cert_file = `</code>	字符串值	自我标识用于客户端身份验证的证书 PEM 文件
<code>`ssl_key_file = `</code>	字符串值	用于为 ssl_cert_file 证书签名的私钥 PEM 文件（可选）
ssl_key_password = None	字符串值	用于解密 ssl_key_file 的密码（如果加密）
ssl_verify_vhost = False	布尔值	默认情况下，SSL 检查服务器证书中的名称与 transport_url 中的主机名匹配。在某些配置中，最好使用虚拟主机名。例如，如果服务器使用 Server Name Indication TLS extension (rfc6066) 来为每个虚拟主机提供证书。如果服务器的 SSL 证书使用虚拟主机名称而不是 DNS 名称，则将 ssl_verify_vhost 设置为 True。
trace = False	布尔值	debug : 将 AMQP 帧转储到 stdout
unicast_address = unicast	字符串值	在发送到特定 RPC/Notification 服务器时，附加到地址前缀。消息总线用来识别发送到单个目的地的消息。

7.1.25. oslo_messaging_kafka

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[oslo_messaging_kafka]` 组下可用的选项。

表 7.24. oslo_messaging_kafka

配置选项 = 默认值	类型	Description
compression_codec = none	字符串值	producer 生成的所有数据的压缩 codec。如果没有设置，则不会使用压缩。请注意，这允许的值取决于 kafka 版本
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟

配置选项 = 默认值	类型	Description
consumer_group = oslo_messaging_consumer	字符串值	Kafka 消费者的组 ID。一个组中的消费者将协调消息消耗
enable_auto_commit = False	布尔值	启用异步消费者提交
kafka_consumer_timeout = 1.0	浮点值	Kafka 用户的默认超时
kafka_max_fetch_bytes = 1048576	整数值	Kafka consumer 的最大获取字节数
max_poll_records = 500	整数值	轮询调用中返回的最大记录数
pool_size = 10	整数值	Kafka Consumers 的池大小
producer_batch_size = 16384	整数值	producer async 发送的批处理大小
producer_batch_timeout = 0.0	浮点值	以秒为单位的 KafkaProducer 批处理延迟上限
sasl_mechanism = PLAIN	字符串值	当安全协议是 SASL 时的机制
security_protocol = PLAINTEXT	字符串值	用于与代理通信的协议
<code>`ssl_cafile = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_client_cert_file = `</code>	字符串值	用于身份验证的客户端证书 PEM 文件。
<code>`ssl_client_key_file = `</code>	字符串值	用于身份验证的客户端密钥 PEM 文件。
<code>`ssl_client_key_password = `</code>	字符串值	用于身份验证的客户端密钥密码文件。

7.1.26. oslo_messaging_notifications

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[oslo_messaging_notifications]` 组下可用的选项。

表 7.25. oslo_messaging_notifications

配置选项 = 默认值	类型	Description
driver = []	多值	处理发送通知的驱动程序。可能的值有 messaging, messagingv2, routing, log, test, noop
retry = -1	整数值	重新发送因为可恢复错误而发送失败的通知消息的最大尝试次数。0 - 不重试, -1 - indefinite
topics = ['notifications']	列表值	用于 OpenStack 通知的 AMQP 主题。
transport_url = None	字符串值	代表用于通知的消息传递驱动程序的 URL。如果没有设置, 我们回退到用于 RPC 的同一配置。

7.1.27. oslo_messaging_rabbit

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[oslo_messaging_rabbit]` 组下可用的选项。

表 7.26. oslo_messaging_rabbit

配置选项 = 默认值	类型	Description
amqp_auto_delete = False	布尔值	AMQP 中的自动删除队列。
amqp_durable_queues = False	布尔值	在 AMQP 中使用持久队列。
direct_mandatory_flag = True	布尔值	(DEPRECATED)为直接发送启用/禁用 RabbitMQ 强制标记。直接发送用作回复, 因此在客户端队列不存在时引发 MessageUndeliverable 异常。MessageUndeliverable 异常将用于循环使发送者恢复。此标志已弃用, 且无法停用此功能
enable_cancel_on_failover = False	布尔值	启用 x-cancel-on-ha-failover 标志, 使 rabbitmq 服务器将取消并通知 consumerswhen 队列停机
heartbeat_in_pthread = False	布尔值	默认情况下, 通过原生 python 线程运行健康检查心跳线程。如果此选项等于 False, 则健康检查 heartbeat 将继承父进程的执行模型。例如, 如果父进程使用 eventlet/greenlet 对 stdlib 进行补丁, 则心跳将通过绿色线程运行。这个选项应该只针对 wsgi 服务设置为 True。
heartbeat_rate = 2	整数值	heartbeat_timeout_threshold 期间的我们检查心跳的频率。
heartbeat_timeout_threshold = 60	整数值	如果 heartbeat 的 keep-alive 失败(0 禁用 heartbeat), 则 Rabbit 代理被视为 down 的秒数。

配置选项 = 默认值	类型	Description
kombu_compression = None	字符串值	EXPERIMENTAL : Possible 值为 : gzip、bz2。如果没有设置压缩, 则不会使用。这个选项可能在以后的版本中不可用。
kombu_failover_strategy = round-robin	字符串值	确定当当前连接到的 RabbitMQ 节点不可用时如何选择下一个 RabbitMQ 节点。仅在配置中提供多个 RabbitMQ 节点时生效。
kombu_missing_consumer_retry_timeout = 60	整数值	在放弃发送其回复前等待缺少的客户端的时间。这个值不应比 <code>rpc_response_timeout</code> 长。
kombu_reconnect_delay = 1.0	浮点值	响应 AMQP 消费者取消通知前等待多久。
rabbit_ha_queues = False	布尔值	尝试在 RabbitMQ 中使用 HA 队列(x-ha-policy: all)。如果更改这个选项, 您必须擦除 RabbitMQ 数据库。在 RabbitMQ 3.0 中, 在声明队列时, x-ha-policy 参数不再控制队列镜像。如果您只想确保所有队列(带有自动生成的名称的队列除外)都在所有节点上进行镜像, 请运行: <code>"rabbitmqctl set_policy HA ^(?!amq\.) gem {"ha-mode": "all"}"</code>
rabbit_interval_max = 30	整数值	RabbitMQ 连接重试的最大间隔。默认值为 30 秒。
rabbit_login_method = AMQPLAIN	字符串值	RabbitMQ 登录方法。
rabbit_qos_prefetch_count = 0	整数值	指定要预抓取的消息数量。设置为零可允许无限消息。
rabbit_retry_backoff = 2	整数值	连接到 RabbitMQ 时重试之间的 backoff 时长。
rabbit_retry_interval = 1	整数值	使用 RabbitMQ 重试连接的频率。
rabbit_transient_queues_ttl = 1800	整数值	正整数代表队列 TTL (x-expires)的持续时间(以秒为单位)。TTL 期间未使用的队列将被自动删除。参数仅影响 reply 和 fanout 队列。
ssl = False	布尔值	通过 SSL 连接。
<code>`ssl_ca_file = `</code>	字符串值	SSL 证书颁发机构文件(仅在启用了 SSL 时评估)。
<code>`ssl_cert_file = `</code>	字符串值	SSL 证书文件(仅在启用了 SSL 时评估)。
<code>`ssl_key_file = `</code>	字符串值	SSL 密钥文件(仅在启用了 SSL 时 valid)。

配置选项 = 默认值	类型	Description
<code>`ssl_version = `</code>	字符串值	要使用的 SSL 版本（仅在启用了 SSL 时评估）。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1_1 和 TLSv1_2 可能在某些发行版中提供。

7.1.28. oslo_middlewre

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[oslo_middlewre]` 组下可用的选项。

表 7.27. oslo_middlewre

配置选项 = 默认值	类型	Description
<code>enable_proxy_headers_parsing = False</code>	布尔值	应用程序是否在代理后面。这决定了中间件是否应该解析标头。
<code>max_request_body_size = 114688</code>	整数值	每个请求的最大正文大小，以字节为单位。
<code>secure_proxy_ssl_header = X-Forwarded-Proto</code>	字符串值	用于确定原始请求协议方案的 HTTP 标头，即使它被 SSL 终止代理隐藏也是如此。

7.1.29. oslo_policy

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[oslo_policy]` 组下可用的选项。

表 7.28. oslo_policy

配置选项 = 默认值	类型	描述
<code>enforce_new_defaults = False</code>	布尔值	这个选项控制在评估策略时是否使用旧的弃用默认值。如果为 True ，则不会评估旧的已弃用的默认值。这意味着，如果旧默认值允许任何现有令牌，但不允许使用新默认值，它将被禁止。我们鼓励启用此标志以及 <code>enforce_scope</code> 标志，以便您可以同时获得新默认值和 <code>scope_type</code> 的好处。
<code>enforce_scope = False</code>	布尔值	此选项控制在评估策略时是否强制执行范围。如果为 True ，则请求中使用的令牌范围将与被强制执行策略的 <code>scope_types</code> 进行比较。如果范围不匹配，则引发 <code>InvalidScope</code> 异常。如果为 False ，则会记录一条信息，告知操作员使用了不匹配的范围调用策略。
<code>policy_default_rule = default</code>	字符串值	默认规则。未找到请求的规则时强制执行。

配置选项 = 默认值	类型	描述
policy_dirs = ['policy.d']	多值	存储策略配置文件的目录。它们可以相对于 config_dir 选项或绝对路径定义的搜索路径中的任何目录。由 policy_file 定义的文件必须存在，才能搜索这些目录。缺少或空目录将被忽略。
policy_file = policy.yaml	字符串值	将角色映射到给定服务权限的文件的相对或绝对路径。必须使用与配置文件设置相关的相对路径。
remote_content_type = application/x-www-form-urlencoded	字符串值	用于发送和接收基于 REST 策略检查的数据的内容类型
remote_ssl_ca_cert_file = None	字符串值	基于 REST 的策略检查的 ca cert 文件的绝对路径
remote_ssl_client_cert_file = None	字符串值	基于 REST 的策略检查的客户端证书的绝对路径
remote_ssl_client_key_file = None	字符串值	绝对路径客户端密钥文件 REST 基于策略检查
remote_ssl_verify_server_cert = False	布尔值	基于 REST 的策略检查的服务器身份验证

7.1.30. policy

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[policy]` 组下可用的选项。

表 7.29. policy

配置选项 = 默认值	类型	Description
driver = sql	字符串值	keystone.policy 命名空间中的策略后端驱动程序的入口点。提供的驱动程序是 规则 （不支持 v3 策略 API 的任何 CRUD 操作）和 sql 。通常，除非您提供了自定义入口点，否则没有设置这个选项的原因。
list_limit = None	整数值	政策集合中返回的最大实体数。

7.1.31. profiler

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[profiler]` 组下可用的选项。

表 7.30. profiler

配置选项 = 默认值	类型	Description
connection_string = messaging://	字符串值	<p>通知程序后端的连接字符串。</p> <p>默认值为 messaging://，它将通知程序设置为 oslo_messaging。</p> <p>可能的值示例：</p> <ul style="list-style-type: none"> ● messaging:// - 使用 oslo_messaging 驱动程序来发送范围。 ● redis://127.0.0.1:6379 - 使用 redis 驱动程序发送 span。 ● MongoDB://127.0.0.1:27017 - 使用 mongodb 驱动程序进行发送范围。 ● Elasticsearch://127.0.0.1:9200 - 使用 elasticsearch 驱动程序来发送 span。 ● Jaeger://127.0.0.1:6831 - 使用 jaeger tracing 作为发送 span 的驱动程序。
enabled = False	布尔值	<p>为此节点上的所有服务启用性能分析。</p> <p>默认值为 False（完全禁用性能分析功能）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用该功能 ● false：禁用该功能。分析无法通过此项目操作启动。如果性能分析由另一个项目触发，则此项目部分将为空。
es_doc_type = notification	字符串值	elasticsearch 中通知索引的文档类型。
es_scroll_size = 10000	整数值	Elasticsearch 批量分割大型请求。此参数定义每个批处理的最大大小（例如：es_scroll_size=10000）。
es_scroll_time = 2m	字符串值	此参数是一个时间值参数（例如：es_scroll_time=2m），指示参与搜索的节点将维护相关资源以继续和支持它。

配置选项 = 默认值	类型	Description
filter_error_trace = False	布尔值	<p>启用包含 error/exception 的过滤器追踪来分隔的位置。</p> <p>默认值为 False。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用包含 error/exception 的过滤器追踪。 ● false：禁用过滤器。
hmac_keys = SECRET_KEY	字符串值	<p>用于加密上下文数据的 secret 密钥，以用于性能分析。</p> <p>此字符串值应该具有以下格式：<key1>[,<key2>,...<keyn>]，其中每个键都是随机字符串。通过 REST API 触发性能分析的用户必须在 REST API 调用的标头中设置其中一个键，使其包含此节点的性能分析结果。</p> <p>"enabled"标志和"hmac_keys"配置选项都应设置为启用性能分析。此外，为了跨所有服务生成正确的性能分析信息，OpenStack 项目之间至少需要一个关键。这样可确保它可用于客户端生成 trace，其中包含所有可能的资源的信息。</p>
sentinel_service_name = mymaster	字符串值	<p>Redis sentinel 使用服务名称来识别 master redis 服务。此参数定义名称（例如：sentinal_service_name=mymaster）。</p>
socket_timeout = 0.1	浮点值	<p>Redis sentinel 在连接上提供了一个 timeout 选项。此参数定义该超时（例如：socket_timeout=0.1）。</p>
trace_sqlalchemy = False	布尔值	<p>在服务中启用 SQL 请求性能分析。</p> <p>默认值为 False (SQL 请求不会被跟踪)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用 SQL 请求性能分析。每个 SQL 查询都将是 trace 的一部分，可以通过花费的时间来分析。 ● false：禁用 SQL 请求性能分析。花费的时间仅显示在更高级别的操作上。无法以这种方式分析单个 SQL 查询。

7.1.32. receipt

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[receipt]` 组下可用的选项。

表 7.31. receipt

配置选项 = 默认值	类型	Description
cache_on_issue = True	布尔值	启用存储签发的接收数据来接收验证缓存，以便首先接收验证不会实际造成完全验证周期。除非启用了全局缓存和接收缓存，否则这个选项无效。
cache_time = 300	整数值	缓存接收创建和验证数据的秒数。除非同时启用了全局和 [receipt] 缓存，否则这不起作用。
caching = True	布尔值	切换到缓存接收创建和验证数据。除非启用了全局缓存，或者禁用了 <code>cache_on_issue</code> ，因为我们只对问题进行缓存接收，否则无效。
expiration = 300	整数值	接收应保持有效的的时间（以秒为单位）。这个值应该始终比较短，因为它代表了用户在使用缺失的 <code>auth</code> 方法重新尝试验证的时间。
provider = fernet	字符串值	keystone.receipt.provider 命名空间中的接收供应商的入口点。receipt 供应商控制接收和验证操作。Keystone 现在仅包含 fernet receipt 提供程序。 fernet receipts 并不需要保留，但要求您运行 keystone-manage fernet_setup (也请参阅 keystone-manage fernet_rotate 命令)。

7.1.33. resource

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[resource]** 组下可用的选项。

表 7.32. resource

配置选项 = 默认值	类型	Description
admin_project_domain_name = None	字符串值	拥有 admin_project_name 的域名。如果保留未设置，则没有 <code>admin</code> 项目。 [resource] admin_project_name 还必须设置为使用这个选项。
admin_project_name = None	字符串值	这是一个特殊的项目，它代表跨服务的云级管理员特权。指向此项目的令牌将包含一个 <code>true</code> is_admin_project 属性，以指示策略特定项目的角色分配应在每个项目中同样应用的系统。如果保留未设置，则没有 <code>admin</code> 项目，因此没有显式方法跨项目角色分配。 [resource] admin_project_domain_name 还必须设置为使用这个选项。
cache_time = None	整数值	缓存资源数据的时间（以秒为单位）。除非启用了全局缓存，否则这无效。

配置选项 = 默认值	类型	Description
caching = True	布尔值	切换资源缓存。除非启用了全局缓存，否则这无效。
domain_name_url_safe = off	字符串值	这控制域名是否仅限于包含 URL-reserved 字符。如果设置为 new ，则尝试使用 URL-unsafe 名称创建或更新域将失败。如果设置为 strict ，则尝试使用 URL-unsafe 域名限定令牌将失败，从而强制更新所有域名为 URL-safe。
driver = sql	字符串值	keystone.resource 命名空间中的资源驱动程序的入口点。keystone 只提供 sql 驱动程序。除非为 keystone 编写专有驱动程序，否则您不需要设置这个选项。
list_limit = None	整数值	在资源集合中返回的最大实体数。
project_name_url_safe = off	字符串值	这控制项目名称是否仅限于包含 URL-reserved 字符。如果设置为 new ，则尝试使用 URL-unsafe 名称创建或更新项目将失败。如果设置为 strict ，则尝试使用 URL-unsafe 项目名称限定令牌将失败，从而强制更新所有项目名称为 URL-safe。

7.1.34. revoke

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[revoke]` 组下可用的选项。

表 7.33. revoke

配置选项 = 默认值	类型	Description
cache_time = 3600	整数值	缓存撤销列表和撤销事件的时间（以秒为单位）。除非同时启用了全局和 <code>[revoke]</code> 缓存，否则这不起作用。
caching = True	布尔值	切换到撤销事件缓存。除非启用了全局缓存，否则这无效。
driver = sql	字符串值	keystone.revoke 命名空间中的令牌撤销后端驱动程序的入口点。Keystone 仅提供 sql 驱动程序，因此不要设置此选项，除非您提供了自定义入口点。
expiration_buffer = 1800	整数值	令牌过期后的秒数，当相应的撤销事件可以从后端清除。

7.1.35. role

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[role]` 组下可用的选项。

表 7.34. role

配置选项 = 默认值	类型	Description
cache_time = None	整数值	缓存角色数据的时间（以秒为单位）。除非同时启用了全局缓存和 [role] 缓存 ，否则这不起作用。
caching = True	布尔值	切换角色缓存。除非启用了全局缓存，否则这无效。在典型的部署中，没有理由禁用此功能。
driver = None	字符串值	keystone.role 命名空间中的角色后端驱动程序的入口点。Keystone 仅提供 sql 驱动程序，因此除非您提供自定义入口点，否则没有更改此设置的原因。
list_limit = None	整数值	角色集合中返回的最大实体数。如果您在部署中有大量离散角色，这对调优非常有用。

7.1.36. saml

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[saml]** 组下可用的选项。

表 7.35. saml

配置选项 = 默认值	类型	Description
assertion_expiration_time = 3600	整数值	确定 keystone 生成的任何 SAML 断言的生命周期，使用 NotOnOrAfter 属性。
certfile = /etc/keystone/ssl/certs/signing_cert.pem	字符串值	用于 SAML 签名的公共证书文件的绝对路径。该值不能包含逗号(、)。
idp_contact_company = Example, Inc.	字符串值	这是身份提供程序的联系人员的公司名称。
idp_contact_email = support@example.com	字符串值	这是身份提供程序联系人员的电子邮件地址。
idp_contact_name = SAML Identity Provider Support	字符串值	这是身份提供程序的联系人员的指定名称。
idp_contact_surname = Support	字符串值	这是身份提供程序的联系人员的姓氏。
idp_contact_telephone = +1 800 555 0100	字符串值	这是身份提供程序的联系人员的电话号码。
idp_contact_type = other	字符串值	这是最适合描述身份提供程序的联系人员的联系人类型。

配置选项 = 默认值	类型	Description
idp_entity_id = None	URI 值	这是生成 SAML 断言时使用的身份提供程序 (keystone) 的唯一实体标识符。生成身份提供程序元数据需要这个值，且必须是 URI（建议使用 URL）。例如： https://keystone.example.com/v3/OS-FEDERATION/saml2/idp 。
idp_lang = en	字符串值	这是身份提供程序的组织使用的语言。
idp_metadata_path = /etc/keystone/saml2_idp_metadata.xml	字符串值	身份提供程序元数据文件的绝对路径。此文件应该使用 keystone-manage saml_idp_metadata 命令生成。通常没有更改此值的原因。
idp_organization_display_name = OpenStack SAML Identity Provider	字符串值	这是要显示的身份提供程序的组织的名称。
idp_organization_name = SAML Identity Provider	字符串值	这是身份提供程序的组织的名称。
idp_organization_url = https://example.com/	URI 值	这是身份提供程序的组织的 URL。这里引用的 URL 对于人类有用。
idp_sso_endpoint = None	URI 值	这是接受 HTTP POST 请求的身份提供程序的单点登录(SSO)服务位置。生成身份提供程序元数据需要值。例如： https://keystone.example.com/v3/OS-FEDERATION/saml2/sso 。
keyfile = /etc/keystone/ssl/private/signing_key.pem	字符串值	用于 SAML 签名的私钥文件的绝对路径。该值不能包含逗号(、)。
relay_state_prefix = ss:mem:	字符串值	生成增强客户端和代理(ECP)断言时使用的 RelayState SAML 属性的前缀。在典型的部署中，没有更改此值的原因。
xmlsec1_binary = xmlsec1	字符串值	用于 XML 签名的二进制文件的名称或绝对路径。虽然只支持 XML 安全库(xmlsec1)，但它可能在您的系统上有一个非标准名称或路径。如果 keystone 无法找到二进制文件，您可能需要安装适当的软件包，使用此选项指定绝对路径，或者调整 keystone 的 PATH 环境变量。

7.1.37. security_compliance

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[security_compliance]` 组下可用的选项。

表 7.36. security_compliance

配置选项 = 默认值	类型	Description
change_password_upon_first_use = False	布尔值	启用此选项要求用户在创建用户时或管理重置时更改密码。在访问任何服务前，受影响的用户必须更改其密码。要忽略特定用户（如服务用户）的这个要求，请通过更新用户 API 为所需的用户设置 options 属性 ignore_change_password_upon_first_use 设置为 True 。此功能默认为禁用。此功能仅适用于 [identity] 驱动程序的 sql 后端。
disable_user_account_days_inactive = None	整数值	在被视为"主动"并自动禁用（锁定）前，用户可以经过验证的最大天数。此功能默认为禁用；将任何值设置为启用它。此功能取决于 [identity] 驱动程序的 sql 后端。当用户超过这个阈值并被视为"inactive"时，HTTP API 中用户的 enabled 属性可能与用户表中 启用的 栏的值不匹配。
lockout_duration = 1800	整数值	超过用户帐户的失败尝试次数上限（由 [security_compliance] lockout_failure_attempts 指定）的最大数量被锁定。除非将 [security_compliance] lockout_failure_attempts 设置为非零值，否则设置此选项将无效。此功能取决于 [identity] 驱动程序的 sql 后端。
lockout_failure_attempts = None	整数值	在由 [security_compliance] lockout_duration 指定的秒数内，用户可以在用户帐户锁定前验证的次数上限。此功能默认为禁用。如果启用了这个功能，并且未设置 [security_compliance] lockout_duration ，则用户可能会无限期锁定，直到用户通过 API 明确启用为止。此功能取决于 [identity] 驱动程序的 sql 后端。
minimum_password_age = 0	整数值	在用户可以更改密码之前必须使用密码的天数。这可防止用户立即更改密码，以擦除密码历史记录并重复使用旧密码。此功能不会阻止管理员手动重置密码。它默认是禁用的，并允许立即更改密码。此功能取决于 [identity] 驱动程序的 sql 后端。注：如果设置了 [security_compliance] password_expires_days ，则此选项的值应小于 password_expires_days 。
password_expires_days = None	整数值	在要求更改密码之前，密码被视为有效的天数。此功能默认为禁用。如果启用，新密码更改将具有过期日期，但现有密码不会受到影响。此功能取决于 [identity] 驱动程序的 sql 后端。
password_regex = None	字符串值	用于验证密码强度要求的正则表达式。默认情况下，正则表达式将匹配任何密码。以下是需要至少 1 个字母、1 个数字且最小长度为 7 个字符的模式示例： ^(?=.*\d)(?=.*[a-zA-Z]).{7,}\$ 此功能取决于 [identity] 驱动程序的 sql 后端。

配置选项 = 默认值	类型	Description
password_regex_description = None	字符串值	以人员使用的语言描述您的密码正则表达式。如果密码无法与正则表达式匹配，则此配置变量的内容将返回到用户，以说明请求的密码不足的原因。
unique_last_password_count = 0	整数值	这将控制在历史记录中保留的之前用户密码迭代的数量，以便强制新创建的密码是唯一的。包含新密码的总数不应大于或等于这个值。将值设为 0（默认值）将禁用此功能。因此，要启用此功能，值必须大于 0。此功能取决于 [identity] 驱动程序的 sql 后端。

7.1.38. shadow_users

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[shadow_users]** 组下可用的选项。

表 7.37. shadow_users

配置选项 = 默认值	类型	Description
driver = sql	字符串值	keystone. identity.shadow_users 命名空间中的 shadow 用户后端驱动程序的入口点。此驱动程序用于对外部管理的身份（通过联邦、LDAP 等）保留本地用户引用。Keystone 仅提供 sql 驱动程序，因此除非您提供自定义入口点，否则没有更改此选项的原因。

7.1.39. token

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[token]** 组下可用的选项。

表 7.38. token

配置选项 = 默认值	类型	Description
allow_expired_window = 172800	整数值	这控制了检索令牌超出内置过期时间的秒数。这允许长时间运行的操作成功。默认为 2 天。
allow_rescope_scoped_token = True	布尔值	这会切换范围性令牌是否可以重新限定到新项目或域，从而防止用户交换任何其他令牌的范围令牌（包括具有默认项目范围）的作用域令牌。这会强制用户对无作用域令牌进行身份验证（稍后会交换具有更具体的范围的令牌的无作用域令牌），或者为每个范围的令牌提供其凭据，以避免完全重复使用。

配置选项 = 默认值	类型	Description
cache_on_issue = True	布尔值	启用将发布的令牌数据存储在令牌验证缓存，以便第一个令牌验证不会实际造成完整的验证周期。除非启用了全局缓存，并且仍然缓存令牌，即使 [token] caching = False ，此选项也不会起作用。弃用自：S *reason:*Keystone 已公开用于缓存令牌的配置选项。在发出令牌时，使用单独的配置选项来缓存令牌，不必要复杂，如果禁用令牌缓存，则误导，因为令牌在签发时默认会预缓存。在发布时预缓存令牌的功能将完全依赖于 "keystone.conf [token] 缓存" 选项。
cache_time = None	整数值	缓存令牌创建和验证数据的秒数。除非同时启用了 global 和 [token] 缓存 ，否则这不起作用。
caching = True	布尔值	切换到缓存令牌创建和验证数据。除非启用了全局缓存，否则这无效。
expiration = 3600	整数值	令牌应保持有效的持续时间（以秒为单位）。显著减少这个值可能会破坏涉及多个服务协调的"长时间运行"操作，并强制用户更频繁地使用 keystone 进行身份验证。显著增加这个值将增加将同时有效的令牌数量。Keystone 令牌也是 bearer 令牌，因此较短的持续时间也会降低被入侵令牌的潜在安全影响。
provider = fernet	字符串值	keystone.token.provider 命名空间中的令牌提供程序的入口点。令牌提供程序控制令牌构造、验证和撤销操作。支持的上游供应商有 fernet 和 jws 。fernet 或 jws 令牌都不需要持久性，且它们都需要额外的设置。如果使用 fernet ，则需要运行 keystone-manage fernet_setup ，这将创建用于加密令牌的对称密钥。如果使用 jws ，则需要使用 SHA-256 哈希算法生成 ECDSA 密钥对，该算法可使用 keystone-manage create_jws_keypair 完成。请注意， fernet 令牌已加密， jws 令牌仅被签名。如果您的部署对用于生成令牌 ID 的有效负载内容有安全要求，请务必考虑这一点。
revoke_by_id = True	布尔值	这会切换支持按令牌标识符撤销各个令牌，因此各种令牌枚举操作（如列出向特定用户发出的所有令牌）。这些操作用于确定要考虑撤销的令牌列表。如果您使用 kvs [revoke] 驱动程序 ，请不要禁用此选项。

7.1.40. tokenless_auth

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[tokenless_auth]** 组下可用的选项。

表 7.39. tokenless_auth

配置选项 = 默认值	类型	Description
issuer_attribute = SSL_CLIENT_I_DN	字符串值	用于将客户端证书签发者传递给 keystone 的 WSGI 环境变量的名称。此属性用作无 X.509 令牌授权的身份提供程序 ID，以及协议查找其对应的映射。在典型的部署中，没有更改此值的原因。
protocol = x509	字符串值	用于代表 X.509 无令牌的授权的联邦协议 ID。这与 [tokenless_auth] issuer_attribute 的值相结合来查找对应的联邦映射。在典型的部署中，没有更改此值的原因。
trusted_issuer = []	多值	可区分名称的列表，用于标识允许使用 X.509 无令牌授权的客户端证书的可信签发者。如果没有该选项，则不允许证书。可区分名称(DN)的值的格式必须用逗号分开，且不包含空格。另外，因为单个 DN 可能会包含逗号，因此该配置选项可能会多次重复，以代表多个值。例如：keystone.conf 会包括两个连续的行来信任两个不同的 DN，如 trusted_issuer = CN=john,OU=keystone,O=openstack 和 trusted_issuer = CN=mary,OU=eng,O=abc 。

7.1.41. totp

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[totp]** 组下可用的选项。

表 7.40. totp

配置选项 = 默认值	类型	Description
included_previous_windows = 1	整数值	在处理 TOTP 传递时要检查的以前的窗口数量。

7.1.42. trust

下表概述了 `/etc/keystone/keystone.conf` 文件中的 **[trust]** 组下可用的选项。

表 7.41. trust

配置选项 = 默认值	类型	Description
allow_redelegation = False	布尔值	允许将授权从一个用户重新定向到另一个用户，并有效地将信任链接在一起。禁用后，信任的 remaining_uses 属性被限制为零。
driver = sql	字符串值	keystone.trust 命名空间中的信任后端驱动程序入口点。Keystone 仅提供 sql 驱动程序，因此除非您提供了自定义入口点，否则没有更改此设置的原因。

配置选项 = 默认值	类型	Description
max_redelegation_count = 3	整数值	授权可以在信任链中从一个用户重新定向到另一个用户的次数上限。对于特定的信任，这个数字可以进一步减少。

7.1.43. unified_limit

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[unified_limit]` 组下可用的选项。

表 7.42. unified_limit

配置选项 = 默认值	类型	Description
cache_time = None	整数值	缓存统一限制数据的时间（以秒为单位）。除非启用了全局缓存和 <code>[unified_limit]</code> 缓存，否则这不起作用。
caching = True	布尔值	切换到统一限制缓存。除非启用了全局缓存，否则这无效。在典型的部署中，没有理由禁用此功能。
driver = sql	字符串值	<code>keystone.unified_limit</code> 命名空间中的统一限制后端驱动程序的入口点。Keystone 仅提供 <code>sql</code> 驱动程序，因此除非您提供自定义入口点，否则没有更改此设置的原因。
enforcement_model = flat	字符串值	验证与项目关联的限制时要使用的强制模型。根据现有限制，执行模型的行为会有所不同，如果模型在运行的部署中切换，这可能会导致向后兼容的更改。
list_limit = None	整数值	在统一限制集合中返回的最大实体数。如果您在部署中有大量统一限制，这对调优非常有用。

7.1.44. wsgi

下表概述了 `/etc/keystone/keystone.conf` 文件中的 `[wsgi]` 组下可用的选项。

表 7.43. wsgi

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
debug_middleware = False	布尔值	如果设置为 true，这会在 Keystone 中启用 oslo 调试中间件。此中间件打印有关请求和响应的许多信息。它可用于获取线（编码）上数据的信息并传递给 WSGI 应用程序管道。此中间件对配置文件的 [DEFAULT] 部分中的 "debug" 设置没有影响，或者将 Keystone 的日志级别设置为 "DEBUG"；它特定于调试 WSGI 数据，并保留 Keystone（特定请求相关的数据）。此选项用于对 web 服务器 (apache、nginx 等) 和 Keystone 之间的请求和响应数据进行内省。此中间件作为中间件链中的第一个元素插入，并显示与线路最接近的数据。警告：不适合在生产环境中使用。此中间件可以和发出敏感/非特权数据。

第 8 章 NEUTRON

以下章节包含有关 `neutron` 服务中的配置选项的信息。

8.1. DHCP_AGENT.INI

本节包含 `/etc/neutron/dhcp_agent.ini` 文件的选项。

8.1.1. DEFAULT

下表概述了 `/etc/neutron/dhcp_agent.ini` 文件中的 `[DEFAULT]` 组下可用的选项。

配置选项 = 默认值	类型	描述
<code>bulk_reload_interval = 0</code>	整数值	重新加载 DHCP 分配之间休眠的时间。这只有在值不是 0 时才会被调用。如果网络在 X 秒内有 N 更新，那么我们将在 X 秒内重新加载端口更改，而不是 N 次。
<code>debug = False</code>	布尔值	如果设置为 <code>true</code> ，则日志级别将设置为 <code>DEBUG</code> ，而不是默认的 <code>INFO</code> 级别。
<code>default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib3.connectionpool=WARN', 'urllib3.connectionpool=WARN', 'websocket=WARN', 'requests.packages.urllib3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WARN', 'routes.middleware=WARN', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INFO']</code>	列表值	以 <code>logger=LEVEL</code> 的形式列出软件包日志级别列表。如果设置了 <code>log_config_append</code> ，则忽略此选项。

配置选项 = 默认值	类型	描述
dhcp_broadcast_reply = False	布尔值	在 DHCP 回复中使用广播。
dhcp_confs = \$state_path/dhcp	字符串值	存储 DHCP 服务器配置文件的位置。
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq	字符串值	用于管理 DHCP 服务器的驱动程序。
dhcp_rebinding_time = 0	整数值	DHCP 重新绑定时间 T2（以秒为单位）。如果设置为 0，则默认为租期时间的 7/8。
dhcp_renewal_time = 0	整数值	DHCP 续订时间 T1（以秒为单位）。如果设置为 0，则默认为 5 个租期时间。
dnsmasq_base_log_dir = None	字符串值	dnsmasq 日志记录的基本日志目录。日志包含 DHCP 和 DNS 日志信息，可用于调试 DHCP 或 DNS 的问题。如果此部分为 null，则禁用 dnsmasq 日志。
<code>`dnsmasq_config_file = `</code>	字符串值	使用此文件覆盖默认的 dnsmasq 设置。
dnsmasq_dns_servers = []	列表值	以逗号分隔的 DNS 服务器列表，这些服务器将用作转发器。
dnsmasq_enable_addr6_list = False	布尔值	当端口在同一子网中有多个 IPv6 地址时，启用 dhcp-host 条目，其中包含地址列表。
dnsmasq_lease_max = 16777216	整数值	限制租期数量，以防止拒绝服务。
dnsmasq_local_resolv = False	布尔值	启用 dnsmasq 服务，在运行 DHCP 代理的主机上通过 DNS 解析器为实例提供名称解析。有效地从 dnsmasq 进程参数中删除 <code>--no-resolv</code> 选项。在 <code>dnsmasq_dns_servers</code> 选项中添加自定义 DNS 解析器可禁用此功能。
enable_isolated_metadata = False	布尔值	DHCP 服务器可协助在隔离的网络上提供元数据支持。将此值设置为 True 将导致 DHCP 服务器将特定的主机路由附加到 DHCP 请求。只有在子网不包含任何路由器端口时，才会激活元数据服务。客户机实例必须配置为通过 DHCP 请求主机路由（选项 121）。当 <code>force_metadata</code> 设为 True 时，此选项没有任何效果。

配置选项 = 默认值	类型	描述
enable_metadata_network = False	布尔值	允许服务来自 CIDR 为 169.254.169.254/16（或更大前缀）的专用元数据访问网络的元数据请求，并连接到虚拟机从中发送元数据的 Neutron 路由器。在这种情况下，DHCP 选项 121 不会在虚拟机中注入，因为它们将通过路由器到达 169.254.169.254。此选项需要 <code>enable_isolated_metadata = True</code> 。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
force_metadata = False	布尔值	在某些情况下，Neutron 路由器没有提供元数据 IP，但 DHCP 服务器可用于提供此信息。设置此值将强制 DHCP 服务器将把特定的主机路由附加到 DHCP 请求。如果设置了这个选项，则会为所有网络激活元数据服务。
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。
interface_driver = None	字符串值	用于管理虚拟接口的驱动程序。
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 <code>log-date-format</code> ）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。

配置选项 = 默认值	类型	描述
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 DEBUG 时，要附加到日志消息的其他数据。 oslo_log.formatters.ContextFormatter 使用
logging_default_format_s tring = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_for mat = %(user) s %(tenant) s %(domain) s % (user_domain) s % (project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
num_sync_threads = 4	整数值	同步过程中使用的线程数量。不应超过服务器上配置的连接池大小。
ovs_integration_bridge = br-int	字符串值	要使用的 Open vSwitch 网桥的名称

配置选项 = 默认值	类型	描述
ovs_use_veth = False	布尔值	使用或不使用 veth 用于 OVS。支持命名空间支持有限的内核（如 RHEL 6.5）和路由器网关端口上的速率限制，只要 ovs_use_veth 设为 True。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
resync_interval = 5	整数值	DHCP 代理将重新同步其状态与 Neutron，以从任何临时通知或 RPC 错误中恢复。间隔是尝试之间的最大秒数。根据触发的事件，可以更频繁地执行 resync。
resync_throttle = 1	整数值	将本地 DHCP 状态和 Neutron 之间的重新同步状态事件数量节流到每个 resync_throttle 秒。throttle 的值引入重新同步状态事件之间的最小间隔。否则重新同步可能会以忙碌循环结束。该值必须小于 resync_interval。
rpc_response_max_timeout = 600	整数值	从 RPC 调用等待响应的最大秒数。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。

配置选项 = 默认值	类型	描述
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append, 则忽略此选项。
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时, 此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 log_file 选项并使用 Linux 平台时, 才有意义。如果设置了 log_config_append, 则忽略此选项。

8.1.2. agent

下表概述了 /etc/neutron/dhcp_agent.ini 文件中的 [agent] 组下可用的选项。

表 8.1. agent

配置选项 = 默认值	类型	Description
availability_zone = nova	字符串值	此节点的可用区
log_agent_heartbeats = False	布尔值	日志代理心跳
report_interval = 30	浮点值	节点报告状态之间的秒数; 应该小于 agent_down_time, 如果节点为一半或小于 agent_down_time。

8.1.3. OVS

下表概述了 /etc/neutron/dhcp_agent.ini 文件中的 [ovs] 组下可用的选项。

表 8.2. OVS

配置选项 = 默认值	类型	Description
bridge_mac_table_size = 50000	整数值	在由 Neutron OVS 代理管理的网桥上学习的最大 MAC 地址数量。合理范围以外的值(10 到 1000,000)可能会根据文档覆盖 Open vSwitch。
igmp_snooping_enable = False	布尔值	为集成网桥启用 IGMP snooping。如果此选项被设置为 True, 则集成桥接中启用了互联网组管理协议 (IGMP) 的支持。将此选项设置为 True 还将启用 Open vSwitch mcast-snooping-disable-flood-unregistered 标志。这个选项将禁用所有端口上取消注册的多播数据包。交换机仅将未注册的多播数据包发送到连接到多播路由器的端口。
ovsdb_connection = tcp:127.0.0.1:6640	字符串值	OVSDB 后端的连接字符串。监控时, 将用于所有 ovsdb 命令和 ovsdb-client

配置选项 = 默认值	类型	Description
ovsdb_debug = False	布尔值	启用 OVSDB 调试日志
ovsdb_timeout = 10	整数值	ovsdb 命令的超时时间（以秒为单位）。如果超时期满，则 ovsdb 命令会失败，并显示 ALARMCLOCK 错误。
ssl_ca_cert_file = None	字符串值	与 OVSDB 交互时使用的证书颁发机构(CA)证书。使用前缀为 ovsdb_connection 的 "ssl:" 时需要
ssl_cert_file = None	字符串值	与 OVSDB 交互时使用的 SSL 证书文件。使用前缀为 ovsdb_connection 的 "ssl:" 时需要
ssl_key_file = None	字符串值	与 OVSDB 交互时使用的 SSL 私钥文件。使用前缀为 ovsdb_connection 的 "ssl:" 时需要

8.2. L3_AGENT.INI

本节包含 `/etc/neutron/l3_agent.ini` 文件的选项。

8.2.1. DEFAULT

下表概述了 `/etc/neutron/l3_agent.ini` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
agent_mode = legacy	字符串值	代理的工作模式。允许模式是 <i>legacy</i> - 这会保留在集中式网络节点上部署 L3 代理的现有行为，以提供 DNAT 和 SNAT 等 L3 服务。如果您不想采用 DVR，请使用此模式。 <i>dvr</i> - 此模式启用 DVR 功能，且必须用于计算主机上运行的 L3 代理。 <i>dvr_snat</i> - 这启用了与 DVR 结合使用的集中式 SNAT 支持。此模式必须用于中央节点上运行的 L3 代理（或在单主机部署中，如 devstack）。 <i>dvr_no_external</i> - 这个模式只为计算主机上运行的 L3 代理启用东/West DVR 路由功能，如 North/South 功能（如 DNAT 和 SNAT）由 <i>dvr_snat</i> 模式运行的集中网络节点提供。当计算主机上没有外部网络连接时，应使用此模式。
api_workers = None	整数值	服务的独立 API worker 进程数量。如果未指定，则默认为等于可用 CPU 数量以获得最佳性能，按潜在的 RAM 使用量上限。
cleanup_on_shutdown = False	布尔值	删除 L3 代理关闭上的所有路由器。对于 L3 HA 路由器，它包括 keepalived 的关闭和状态更改监控器。注意：在停止或重启 L3 代理时，设置为 True 可能会影响数据平面。

配置选项 = 默认值	类型	Description
debug = False	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpidd=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。 如果设置了 log_config_append，则忽略此选项。
enable_metadata_proxy = True	布尔值	允许运行元数据代理。
external_ingress_mark = 0x2	字符串值	用于标记来自外部网络的 ingress 的 iptables mangle 标记。此标记将屏蔽为 0xffff，以便只使用低 16 位。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
ha_confs_path = \$state_path/ha_confs	字符串值	保存 keepalived 配置文件的位置
ha_keepalived_state_change_server_threads = <based on operating system>	整数值	keepalived 服务器连接请求的并发线程数量。更多线程在代理节点上创建更高的 CPU 负载。

配置选项 = 默认值	类型	Description
ha_vrrp_advert_int = 2	整数值	广告间隔（以秒为单位）
ha_vrrp_auth_password = None	字符串值	VRRP 验证密码
ha_vrrp_auth_type = PASS	字符串值	VRRP 验证类型
ha_vrrp_garp_master_delay = 5	整数值	在 MASTER 时收到较低优先级 advert 后第二组 gratuitous ARP 的延时。注意：此配置选项仅适用于 OSP13 和 OSP16。将来的版本将实施模板表单以提供"keepalived"配置。
ha_vrrp_garp_master_repeats = 5	整数值	过渡到 MASTER 后要发送的 gratuitous ARP 消息数量。注意：此配置选项仅适用于 OSP13 和 OSP16。将来的版本将实施模板表单以提供"keepalived"配置。
ha_vrrp_health_check_interval = 0	整数值	VRRP 健康检查间隔（以秒为单位）。值 > 0 启用 VRRP 健康检查。将它设置为 0 可禁用 VRRP 健康检查。推荐的值为 5。这将导致 ping 发送到网关 IP 地址 - 要求在网关上启用 ICMP_ECHO_REQUEST。如果网关失败，则所有路由器都将报告为主要机制，并且以轮循方式重复主选举，直到其中一个路由器恢复网关连接。
handle_internal_only_routers = True	布尔值	表示此 L3 代理还应处理没有配置外部网络网关的路由器。这个选项应该仅为 Neutron 部署中的单一代理为 True，如果所有路由器都必须具有外部网络网关，则所有代理可能是 False。
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。
interface_driver = None	字符串值	用于管理虚拟接口的驱动程序。
<code>`ipv6_gateway = `</code>	字符串值	对于 IPv6，用于外部网关的网络不需要关联的子网，因为可以使用自动分配的本地链路地址(LLA)。但是，需要一个 IPv6 网关地址，用作默认路由的下一跃点。如果此处未配置 IPv6 网关地址，（然后，将 neutron 路由器配置为从路由器公告(RA)获取其默认路由），则上游路由器还必须配置为发送这些 RA。在配置时，ipv6_gateway 应该是上游路由器上接口的 LLA。如果需要使用全局唯一地址(GUA)的下一跃点，需要通过分配给网络的子网来完成，而不是通过此参数。

配置选项 = 默认值	类型	Description
keepalived_use_no_track = True	布尔值	如果使用没有支持 "no_track" 选项的 keepalived, 则应将其设置为 False。keepalived 2.x 中引入了对这个选项的支持
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情, 请查看 Python 日志记录模块文档。请注意, 当使用日志记录配置文件时, 所有日志记录配置都会在配置文件中设置, 其他日志记录配置选项将被忽略 (如 log-date-format)。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 %(asctime) s 的格式字符串。默认: %(default) s。如果设置了 log_config_append, 则忽略此选项。
log-dir = None	字符串值	(可选) 用于相对 log_file 路径的基础目录。如果设置了 log_config_append, 则忽略此选项。
log-file = None	字符串值	(可选) 要将日志输出发送到的日志文件名称。如果没有设置默认值, 日志记录将进入由 use_stderr 定义的 stderr。如果设置了 log_config_append, 则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 log_rotation_type 设置为 "interval", 否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时, 将使用上次文件更改的时间 (或服务启动的时间)。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。oslo_log.formatters.ContextFormatter 使用
logging_debug_format_suffix = %(funcName)s %(pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 DEBUG 时, 要附加到日志消息的其他数据。oslo_log.formatters.ContextFormatter 使用

配置选项 = 默认值	类型	Description
logging_default_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_format = %(user) s %(tenant) s %(domain) s % (user_domain) s % (project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
max_rtr_adv_interval = 100	整数值	radvd.conf 的 MaxRtrAdvInterval 设置
metadata_access_mark = 0x1	字符串值	用于标记元数据有效请求的 iptables mangle 标记。此标记将屏蔽为 0xffff，以便只使用低 16 位。
metadata_port = 9697	端口值	Neutron 元数据命名空间代理使用的 TCP 端口。
min_rtr_adv_interval = 30	整数值	MinRtrAdvInterval setting for radvd.conf
ovs_integration_bridge = br-int	字符串值	要使用的 Open vSwitch 网桥的名称
ovs_use_veth = False	布尔值	使用或不使用 veth 用于 OVS。支持命名空间支持有限的内核（如 RHEL 6.5）和路由器网关端口上的速率限制，只要 ovs_use_veth 设为 True。
pd_confs = \$state_path/pd	字符串值	存储 IPv6 PD 文件的位置。
periodic_fuzzy_delay = 5	整数值	启动定期任务调度程序时随机延迟的秒数，以减少时间戳。（通过设置为 0 来禁用）

配置选项 = 默认值	类型	Description
periodic_interval = 40	整数值	运行定期任务之间的秒。
prefix_delegation_driver = dibbler	字符串值	用于 ipv6 前缀长度的驱动程序。这需要是 neutron.agent.linux.pd_drivers 命名空间中定义的入口点。有关 neutron 源中包含的入口点，请参阅 setup.cfg。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
ra_confs = \$state_path/ra	字符串值	存储 IPv6 RA 配置文件的位置
<code>`radvd_user = `</code>	字符串值	传递给 radvd 的用户名，用于丢弃 root 特权，并将用户 ID 更改为 username 的主组 ID。如果没有指定用户（默认为），将传递执行 L3 代理的用户。如果指定了 "root"，因为 radvd 会生成为 root，则不会传递 "username" 参数。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
rpc_response_max_timeout = 600	整数值	从 RPC 调用等待响应的最大秒数。
rpc_state_report_workers = 1	整数值	专用于状态报告队列的 RPC worker 进程数量。
rpc_workers = None	整数值	服务的 RPC worker 进程数。如果没有指定，则默认等于 API worker 的数量。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。

配置选项 = 默认值	类型	Description
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。
vendor_pen = 8888	字符串值	RFC3315-EN 要求使用厂商的 Registered Private Enterprise Number 的十进制值。
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 log_file 选项并使用 Linux 平台时，才有意义。如果设置了 log_config_append，则忽略此选项。

8.2.2. agent

下表概述了 /etc/neutron/l3_agent.ini 文件中的 [agent] 组下可用的选项。

表 8.3. agent

配置选项 = 默认值	类型	Description
availability_zone = nova	字符串值	此节点的可用区
extensions = []	列表值	要使用的扩展列表
log_agent_heartbeats = False	布尔值	日志代理心跳
report_interval = 30	浮点值	节点报告状态之间的秒数；应该小于 agent_down_time，如果节点为一半或小于 agent_down_time。

8.2.3. network_log

下表概述了 /etc/neutron/l3_agent.ini 文件中的 [network_log] 组下可用的选项。

表 8.4. network_log

配置选项 = 默认值	类型	Description
burst_limit = 25	整数值	每个 rate_limit 的最大数据包数量。
local_output_log_base = None	字符串值	输出代理侧的 logfile 路径，默认 syslog 文件。
rate_limit = 100	整数值	每秒数据包日志记录数上限。

8.2.4. OVS

下表概述了 `/etc/neutron/l3_agent.ini` 文件中的 `[ovs]` 组下可用的选项。

表 8.5. OVS

配置选项 = 默认值	类型	Description
bridge_mac_table_size = 50000	整数值	在由 Neutron OVS 代理管理的网桥上学习的最大 MAC 地址数量。合理范围以外的值(10 到 1000,000)可能会根据文档覆盖 Open vSwitch。
igmp_snooping_enable = False	布尔值	为集成网桥启用 IGMP snooping。如果此选项被设置为 True，则集成桥接中启用了互联网组管理协议 (IGMP) 的支持。将此选项设置为 True 还将启用 Open vSwitch mcast-snooping-disable-flood-unregistered 标志。这个选项将禁用所有端口上取消注册的多播数据包。交换机仅将未注册的多播数据包发送到连接到多播路由器的端口。
ovsdb_connection = tcp:127.0.0.1:6640	字符串值	OVSDB 后端的连接字符串。监控时，将用于所有 ovsdb 命令和 ovsdb-client
ovsdb_debug = False	布尔值	启用 OVSDB 调试日志
ovsdb_timeout = 10	整数值	ovsdb 命令的超时时间（以秒为单位）。如果超时到期，则 ovsdb 命令会失败，并显示 ALARMCLOCK 错误。
ssl_ca_cert_file = None	字符串值	与 OVSDB 交互时使用的证书颁发机构(CA)证书。使用前缀为 ovsdb_connection 的 "ssl:" 时需要
ssl_cert_file = None	字符串值	与 OVSDB 交互时使用的 SSL 证书文件。使用前缀为 ovsdb_connection 的 "ssl:" 时需要
ssl_key_file = None	字符串值	与 OVSDB 交互时使用的 SSL 私钥文件。使用前缀为 ovsdb_connection 的 "ssl:" 时需要

8.3. LINUXBRIDGE_AGENT.INI

本节包含 `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` 文件的选项。

8.3.1. DEFAULT

下表概述了 `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
debug = False	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。
default_log_levels = ['amqp=WARN', 'amqplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib3.connectionpool=WARN', , 'urllib3.connectionpool=WARN', 'websocket=WARN', 'requests.packages.urllib3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WARN', 'routes.middleware=WARN', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INFO']	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 log_config_append，则忽略此选项。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。

配置选项 = 默认值	类型	Description
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 log-date-format）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 <code>DEBUG</code> 时，要附加到日志消息的其他数据。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_default_format_s tring = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用

配置选项 = 默认值	类型	Description
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_format = %(user) s %(tenant) s %(domain) s %(user_domain) s %(project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
rpc_response_max_timeout = 600	整数值	从 RPC 调用等待响应的最大秒数。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。

配置选项 = 默认值	类型	Description
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 log_file 选项并使用 Linux 平台时，才有意义。如果设置了 log_config_append，则忽略此选项。

8.3.2. agent

下表概述了 `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` 文件中的 `[agent]` 组下可用的选项。

表 8.6. agent

配置选项 = 默认值	类型	Description
dscp = None	整数值	在隧道封装过程中用于外部标头的 DSCP 值。
dscp_inherit = False	布尔值	如果设置为 True，则隧道接口的 DSCP 值会被覆盖，并设置为 inherit。然后，内部标头的 DSCP 值被复制到 outer 标头。
extensions = []	列表值	要使用的扩展列表
polling_interval = 2	整数值	代理在轮询本地设备更改之间等待的时间（以秒为单位）。
quitting_rpc_timeout = 10	整数值	为代理接收 SIGTERM 后新的 rpc 调用设置新的超时时间（以秒为单位）。如果值设为 0，则 rpc timeout 不会更改

8.3.3. linux_bridge

下表概述了 `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` 文件中的 `[linux_bridge]` 组下可用的选项。

表 8.7. linux_bridge

配置选项 = 默认值	类型	Description
bridge_mappings = []	列表值	<physical_network> 列表<physical_bridge>
physical_interface_mappings = []	列表值	以逗号分隔的 <physical_network> : <physical_interface> 将物理网络名称映射到代理的特定物理网络接口，以用于扁平化和 VLAN 网络。服务器上 network_vlan_ranges 中列出的所有物理网络应具有到每个代理上相应接口的映射。

8.3.4. network_log

下表概述了 `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` 文件中的 **[network_log]** 组下可用的选项。

表 8.8. network_log

配置选项 = 默认值	类型	Description
burst_limit = 25	整数值	每个 rate_limit 的最大数据包数量。
local_output_log_base = None	字符串值	输出代理侧的 logfile 路径，默认 syslog 文件。
rate_limit = 100	整数值	每秒数据包日志记录数上限。

8.3.5. securitygroup

下表概述了 `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` 文件中的 **[securitygroup]** 组下可用的选项。

表 8.9. securitygroup

配置选项 = 默认值	类型	Description
enable_ipset = True	布尔值	使用 ipset 来加快基于 iptables 的安全组。启用 ipset 支持要求在 L2 代理节点上安装 ipset。
enable_security_group = True	布尔值	控制服务器中是否启用了 neutron 安全组 API。在不使用安全组或使用 nova 安全组 API 时，它应当为 false。
firewall_driver = None	字符串值	L2 代理中的安全组防火墙的驱动程序
permitted_ethertypes = []	列表值	允许以逗号分隔的 ethertypes 列表，以十六进制表示十六进制（从 "0x" 开始）。例如，"0x4008" 以允许 InfiniBand。

8.3.6. vxlan

下表概述了 `/etc/neutron/plugins/ml2/linuxbridge_agent.ini` 文件中的 `[vxlan]` 组下可用的选项。

表 8.10. vxlan

配置选项 = 默认值	类型	Description
arp_responder = False	布尔值	启用本地 ARP 响应器，其提供本地响应而不是在覆盖中执行 ARP 广播。启用本地 ARP 响应器与允许地址对扩展不完全兼容。
enable_vxlan = True	布尔值	在代理上启用 VXLAN。在使用 linuxbridge 机制驱动程序由 ml2 插件管理代理时可以启用
l2_population = False	布尔值	扩展，与 ml2 插件的 l2population 机制驱动程序一起使用。它可让插件填充 VXLAN 转发表。
local_ip = None	IP 地址值	本地覆盖（隧道）网络端点的 IP 地址。使用位于其中一个主机网络接口的 IPv4 或 IPv6 地址。这个值的 IP 版本必须与 neutron 服务器节点上的 ML2 插件配置文件中的 <code>overlay_ip_version</code> 选项值匹配。
multicast_ranges = []	列表值	可选以逗号分隔的 <code><multicast address>:<vni_min>:<vni_max></code> triples 列表，描述了如何根据其 VNI ID 为 VXLAN 分配多播地址。
ToS = None	整数值	TOS 用于 vxlan 接口协议数据包。这个选项已弃用，现在使用 AGENT 部分中的 <code>dscp</code> 选项替代，并将在以后的版本中删除。要将 TOS 值转换为 DSCP，请划分 4。
tll = None	整数值	vxlan 接口协议数据包的 TTL。
udp_dstport = None	端口值	用于 VXLAN 通信的 UDP 端口。默认情况下，Linux 内核不使用 IANA 分配的标准值，因此如果您想要使用它，此选项必须设置为 4789。默认情况下不设置它，因为向后兼容。
udp_srcport_max = 0	端口值	用于 VXLAN 通信的最大 UDP 源端口范围。
udp_srcport_min = 0	端口值	用于 VXLAN 通信的最小 UDP 源端口范围。
vxlan_group = 224.0.0.1	字符串值	vxlan 接口的多播组。可以使用 CIDR 表示法指定组地址范围。指定范围允许不同的 VNI 使用不同的组地址，减少或消除到隧道端点的伪装广播流量。要为每个可能(24-bit) VNI 保留一个唯一的组，请使用 /8，如 239.0.0.0/8。在所有代理中，此设置都必须相同。

8.4. METADATA_AGENT.INI

本节包含 `/etc/neutron/metadata_agent.ini` 文件的选项。

8.4.1. DEFAULT

下表概述了 `/etc/neutron/metadata_agent.ini` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
auth_ca_cert = None	字符串值	ssl 的证书颁发机构公钥(CA 证书)文件
debug = False	布尔值	如果设置为 true, 则日志级别将设置为 DEBUG, 而不是默认的 INFO 级别。
default_log_levels = <code>['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpidd=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' ', 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']</code>	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 <code>log_config_append</code> , 则忽略此选项。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。

配置选项 = 默认值	类型	Description
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 log-date-format）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	字符串值	使用上下文格式化用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 <code>DEBUG</code> 时，要附加到日志消息的其他数据。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_default_format_s tring = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s	字符串值	上下文未定义时用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用

配置选项 = 默认值	类型	Description
logging_exception_prefix = <code>%(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s</code>	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_format = <code>%(user) s %(tenant) s %(domain) s %(user_domain) s %(project_domain) s</code>	字符串值	定义 logging_context_format_string 中使用的 <code>%(user_identity) s</code> 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
metadata_backlog = 4096	整数值	配置元数据服务器套接字的积压请求数
<code>metadata_proxy_group = `</code>	字符串值	在初始化后运行元数据代理的组(gid 或名称)（如果为空：代理有效组）。
<code>metadata_proxy_shared_secret = `</code>	字符串值	在代理元数据请求时，Neutron 使用共享 secret 为 Instance-ID 标头签名，以防止欺骗。您可以为 secret 选择任何字符串，但它必须与 Nova 元数据服务器使用的配置中匹配。注意：Nova 使用相同的配置密钥，但在 [neutron] 部分中。
metadata_proxy_socket = <code>\$state_path/metadata_proxy</code>	字符串值	元数据代理 UNIX 域套接字的位置。
metadata_proxy_socket_mode = <code>deduce</code>	字符串值	元数据代理 UNIX 域套接字模式，允许 4 个值： <code>deduce</code> - 从 metadata_proxy_user/group 值的分离模式； <code>user</code> - 将元数据代理套接字模式设置为 0o644，以便在 metadata_proxy_user 是代理有效用户或 root 用户时使用； <code>group</code> - 将元数据代理套接字模式设置为 0o664，以便在 metadata_proxy_group 是有效的代理组或用户时使用； <code>all</code> - 将原始代理套接字模式设置为 0o666，在所有其他情况下使用。
<code>metadata_proxy_user = `</code>	字符串值	用户(uid 或名称)在其初始化后运行元数据代理（如果为空：代理有效的用户）。
metadata_workers = <code><based on operating system></code>	整数值	元数据服务器的单独 worker 进程数量（与 ML2/OVN 一起使用时默认为 2，与其他后端驱动程序的 CPU 数量有一半）

配置选项 = 默认值	类型	Description
<code>`nova_client_cert = `</code>	字符串值	nova 元数据 api 服务器的客户端证书。
<code>`nova_client_priv_key = `</code>	字符串值	客户端证书的私钥。
nova_metadata_host = 127.0.0.1	主机地址值	Nova 元数据服务器的 IP 地址或 DNS 名称。
nova_metadata_insecure = False	布尔值	允许对 nova 元数据执行不安全的 SSL (https) 请求
nova_metadata_port = 8775	端口值	Nova 元数据服务器使用的 TCP 端口。
nova_metadata_protocol = http	字符串值	协议来访问 nova 元数据、http 或 https
publish_errors = False	布尔值	启用或禁用错误事件的发布。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
rpc_response_max_timeout = 600	整数值	从 RPC 调用等待响应的最大秒数。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。

配置选项 = 默认值	类型	Description
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 log_file 选项并使用 Linux 平台时，才有意义。如果设置了 log_config_append，则忽略此选项。

8.4.2. agent

下表概述了 `/etc/neutron/metadata_agent.ini` 文件中的 `[agent]` 组下可用的选项。

表 8.11. agent

配置选项 = 默认值	类型	Description
log_agent_heartbeats = False	布尔值	日志代理心跳
report_interval = 30	浮点值	节点报告状态之间的秒数；应该小于 agent_down_time，如果节点为一半或小于 agent_down_time。

8.4.3. 缓存

下表概述了 `/etc/neutron/metadata_agent.ini` 文件中的 `[cache]` 组下可用的选项。

表 8.12. 缓存

配置选项 = 默认值	类型	Description
backend = dogpile.cache.null	字符串值	缓存后端模块。对于具有数百个线程服务器的基于 eventlet 的环境，建议使用带有池 (oslo_cache.memcache_pool) 的 Memcache。对于小于 100 个线程服务器的环境，建议使用 Memcached (dogpile.cache.memcached) 或 Redis (dogpile.cache.redis)。具有单一服务器实例的测试环境可以使用 dogpile.cache.memory 后端。

配置选项 = 默认值	类型	Description
backend_argument = []	多值	为 backend 模块提供的参数。指定每参数传递到 dogpile.cache 后端的一次这个选项。示例格式："<argname>:<value>"。
config_prefix = cache.oslo	字符串值	为缓存区域构建配置字典的前缀。除非存在另一个具有相同配置名称的 dogpile.cache 区域，否则这不应更改。
dead_timeout = 60	浮点值	尝试在 HashClient 内部机制中重新添加节点前的时间（以秒为单位）。
debug_cache_backend = False	布尔值	从缓存后端进行额外的调试（缓存键、get/set/delete/etc 调用）。只有在您需要使用键/值查看特定的 cache-backend get/set/delete 调用时才有用。通常，这应该设为 false。
enable_retry_client = False	布尔值	启用重试客户端机制来处理失败。这些机制可用于嵌套所有的 pymemcache 客户端。打包程序允许您定义有多少尝试进行，以及 attempts 之间等待的时间。
enable_socket_keepalive = False	布尔值	dogpile 的 pymemcache 后端的套接字 keepalive 的全局切换
enabled = False	布尔值	缓存的全局切换。
expiration_time = 600	整数值	dogpile.cache 区域中的任何缓存项的默认 TTL（以秒为单位）。这适用于任何未为其定义显式缓存过期时间的缓存方法。
hashclient_retry_attempts = 2	整数值	在客户端标记为死前应尝试的次数，并在 HashClient 的内部机制中从池中删除。
hashclient_retry_delay = 1	浮点值	在 HashClient 内部机制中重试尝试之间应经过的时间（以秒为单位）。
memcache_dead_retry = 300	整数值	在再次尝试前，memcached 服务器被视为 dead 的秒数。（dogpile.cache.memcache 和 oslo_cache.memcache_pool 后端）。
memcache_pool_connection_get_timeout = 10	整数值	操作将等待的秒数，以获取 memcache 客户端连接。
memcache_pool_flush_on_reconnect = False	布尔值	如果 memcache 在重新连接时清除，则全局切换。（仅oslo_cache.memcache_pool 后端）。

配置选项 = 默认值	类型	Description
memcache_pool_maxsize = 10	整数值	每个 memcached 服务器的最大打开连接数。 (oslo_cache.memcache_pool backend)。
memcache_pool_unused_timeout = 60	整数值	到 memcached 的连接在池关闭前没有被使用的秒数。 (仅oslo_cache.memcache_pool 后端)。
memcache_servers = ['localhost:11211']	列表值	memcache 服务器格式为 "host:port"。 (dogpile.cache.memcached 和 oslo_cache.memcache_pool 后端)。如果给定主机引用 IPv6 或给定域, 则您应该为给定地址系列(inet6) (inet6) (inet6) (inet6[::1]:11211,inet6:[fd12:3456:789a:1::1]:11211,inet6:[controller-0.internalapi]:11211)作为前缀。如果未提供地址系列, 则使用的默认地址系列将是 inet (对应于 IPv4)
memcache_socket_timeout = 1.0	浮点值	每次对服务器调用的超时时间 (以秒为单位)。 (dogpile.cache.memcache 和 oslo_cache.memcache_pool 后端)。
proxies = []	列表值	用于导入的代理类会影响 dogpile.cache 后端功能的方式。有关 change-backend-behavior 的 dogpile.cache 文档。
retry_attempts = 2	整数值	在失败前尝试操作的次数。
retry_delay = 0	浮点值	每次尝试之间休眠的秒数。
socket_keepalive_count = 1	整数值	在丢弃连接前, 保持 TCP 的最大 keepalive 探测 TCP 数量。应该是大于零的正整数。
socket_keepalive_idle = 1	整数值	在 TCP 开始发送 keepalive 探测前, 连接需要保持闲置的时间 (以秒为单位)。应该是大于零的正整数。
socket_keepalive_interval = 1	整数值	各个 keepalive 探测之间的时间 (以秒为单位)。应该是大于零的正整数。
tls_allowed_ciphers = None	字符串值	为使用 TLS 上下文创建的套接字设置可用密码。它应该是 OpenSSL 密码列表格式的字符串。如果没有指定, 则所有启用了 OpenSSL 的密码都可用。
tls_cafile = None	字符串值	以 PEM 格式串联的 CA 证书文件的路径, 以建立缓存服务器的真实性。如果 <code>tls_enabled</code> 是 <code>False</code> , 这个选项将被忽略。
tls_certfile = None	字符串值	到 PEM 格式的单个文件的路径, 其中包含客户端证书以及建立证书真实性所需的任意数量的 CA 证书。只有在需要客户端身份验证时才需要此文件。如果 <code>tls_enabled</code> 是 <code>False</code> , 这个选项将被忽略。

配置选项 = 默认值	类型	Description
tls_enabled = False	布尔值	与缓存服务器组合时，TLS 使用的全局切换。
tls_keyfile = None	字符串值	到包含客户端私钥的单个文件的路径。其他私钥将从 tls_certfile 中指定的文件获取。如果 tls_enabled 是 False，这个选项将被忽略。

8.5. METERING_AGENT.INI

本节包含 `/etc/neutron/metering_agent.ini` 文件的选项。

8.5.1. DEFAULT

下表概述了 `/etc/neutron/metering_agent.ini` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
debug = False	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。

配置选项 = 默认值	类型	Description
default_log_levels = ['amqp=WARN', 'amqp-lib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。 如果设置了 log_config_append，则忽略此选项。
driver = neutron.services.meterin g.drivers.noop.noop_driv er.NoopMeteringDriver	字符串值	Metering 驱动程序
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
granular_traffic_data = False	布尔值	定义 metering 代理驱动程序是否应该以粒度方式呈现流量数据，而不是对标签分配到的所有项目和路由器的流量数据进行分组。默认值为 False ，用于向后兼容。
`instance_format = [instance: %(uuid)s]`	字符串值	通过日志消息传递的实例格式。
`instance_uuid_format = [instance: %(uuid)s]`	字符串值	通过日志消息传递的实例 UUID 的格式。
interface_driver = None	字符串值	用于管理虚拟接口的驱动程序。

配置选项 = 默认值	类型	Description
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 log-date-format）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 <code>DEBUG</code> 时，要附加到日志消息的其他数据。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_default_format_s tring = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用

配置选项 = 默认值	类型	Description
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_format = %(user) s %(tenant) s %(domain) s % (user_domain) s % (project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size", 则忽略这个选项。
measure_interval = 30	整数值	两个 metering 测量之间的间隔
ovs_integration_bridge = br-int	字符串值	要使用的 Open vSwitch 网桥的名称
ovs_use_veth = False	布尔值	使用或不使用 veth 用于 OVS。支持命名空间支持有限的内核（如 RHEL 6.5）和路由器网关端口上的速率限制，只要 ovs_use_veth 设为 True。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
report_interval = 300	整数值	两个 metering 报告之间的间隔
rpc_response_max_timeout = 600	整数值	从 RPC 调用等待响应的最大秒数。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append, 则忽略此选项。

配置选项 = 默认值	类型	Description
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 log_file 选项并使用 Linux 平台时，才有意义。如果设置了 log_config_append，则忽略此选项。

8.5.2. agent

下表概述了 /etc/neutron/metering_agent.ini 文件中的 [agent] 组下可用的选项。

表 8.13. agent

配置选项 = 默认值	类型	Description
log_agent_heartbeats = False	布尔值	日志代理心跳
report_interval = 30	浮点值	节点报告状态之间的秒数；应该小于 agent_down_time，如果节点为一半或小于 agent_down_time。

8.5.3. OVS

下表概述了 /etc/neutron/metering_agent.ini 文件中的 [ovs] 组下可用的选项。

表 8.14. OVS

配置选项 = 默认值	类型	Description
bridge_mac_table_size = 50000	整数值	在由 Neutron OVS 代理管理的网桥上学习的最大 MAC 地址数量。合理范围以外的值(10 到 1000,000)可能会根据文档覆盖 Open vSwitch。
igmp_snooping_enable = False	布尔值	为集成网桥启用 IGMP snooping。如果此选项被设置为 True, 则集成桥接中启用了互联网组管理协议 (IGMP)的支持。将此选项设置为 True 还将启用 Open vSwitch mcast-snooping-disable-flood-unregistered 标志。这个选项将禁用所有端口上取消注册的多播数据包。交换机仅将未注册的多播数据包发送到连接到多播路由器的端口。
ovsdb_connection = tcp:127.0.0.1:6640	字符串值	OVSDB 后端的连接字符串。监控时, 将用于所有 ovsdb 命令和 ovsdb-client
ovsdb_debug = False	布尔值	启用 OVSDB 调试日志
ovsdb_timeout = 10	整数值	ovsdb 命令的超时时间 (以秒为单位)。如果超时到期, 则 ovsdb 命令会失败, 并显示 ALARMCLOCK 错误。
ssl_ca_cert_file = None	字符串值	与 OVSDB 交互时使用的证书颁发机构(CA)证书。使用前缀为 ovsdb_connection 的 "ssl:" 时需要
ssl_cert_file = None	字符串值	与 OVSDB 交互时使用的 SSL 证书文件。使用前缀为 ovsdb_connection 的 "ssl:" 时需要
ssl_key_file = None	字符串值	与 OVSDB 交互时使用的 SSL 私钥文件。使用前缀为 ovsdb_connection 的 "ssl:" 时需要

8.6. ML2_CONF.INI

本节包含 `/etc/neutron/plugins/ml2/ml2_conf.ini` 文件的选项。

8.6.1. DEFAULT

下表概述了 `/etc/neutron/plugins/ml2/ml2_conf.ini` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
debug = False	布尔值	如果设置为 true, 则日志级别将设置为 DEBUG, 而不是默认的 INFO 级别。

配置选项 = 默认值	类型	Description
default_log_levels = <code>['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpid=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']</code>	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 log_config_append，则忽略此选项。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 log-date-format）。

配置选项 = 默认值	类型	Description
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	(可选) 用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	(可选) 要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间(或服务启动的时间)。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime)s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 <code>DEBUG</code> 时，要附加到日志消息的其他数据。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_default_format_s tring = %(asctime)s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 <code>oslo_log.formatters.ContextFormatter</code> 使用

配置选项 = 默认值	类型	Description
logging_user_identity_format = %(user) s %(tenant) s %(domain) s %(user_domain) s %(project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。

配置选项 = 默认值	类型	Description
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 <code>log_file</code> 选项并使用 Linux 平台时，才有意义。如果设置了 <code>log_config_append</code> ，则忽略此选项。

8.6.2. ml2

下表概述了 `/etc/neutron/plugins/ml2/ml2_conf.ini` 文件中的 `[ml2]` 组下可用的选项。

表 8.15. ml2

配置选项 = 默认值	类型	Description
extension_drivers = []	列表值	要从 <code>neutron.ml2.extension_drivers</code> 命名空间中载入的扩展驱动程序入口点列表。例如： <code>extension_drivers = port_security,qos</code>
external_network_type = None	字符串值	如果没有指定供应商属性，则外部网络的默认网络类型。默认情况下，它是 <code>None</code> ，这意味着如果在创建外部网络时未指定提供程序属性，则它们将与租户网络相同。 <code>external_network_type</code> 配置选项允许的值取决于 <code>type_drivers</code> 配置选项中配置的网络类型值。
mechanism_drivers = []	列表值	要从 <code>neutron.ml2.mechanism_drivers</code> 命名空间中载入的网络机制驱动程序入口点的有序列表。
overlay_ip_version = 4	整数值	所有覆盖（隧道）网络端点的 IP 版本。4 代表 IPv4，6 代表 IPv6。
path_mtu = 0	整数值	使用 <code>overlay/tunnel</code> 协议时，可以遍历底层物理网络基础架构的最大大小，而无需碎片。此选项允许指定与默认的 <code>global_physnet_mtu</code> 值不同的物理网络 MTU 值。
physical_network_mtus = []	列表值	物理网络到 MTU 值的映射列表。映射的格式是 <code><physnet>:<mtu val></code> 。此映射允许指定与默认的 <code>global_physnet_mtu</code> 值不同的物理网络 MTU 值。
tenant_network_types = ['local']	列表值	排序 <code>network_types</code> 列表，以分配为租户网络。默认值为 <code>local</code> ，可用于单箱测试，但主机之间不提供连接。
type_drivers = ['local', 'flat', 'vlan', 'gre', 'vxlan', 'geneve']	列表值	要从 <code>neutron.ml2.type_drivers</code> 命名空间中载入的网络类型入口点列表。

8.6.3. ml2_type_flat

下表概述了 `/etc/neutron/plugins/ml2/ml2_conf.ini` 文件中的 `[ml2_type_flat]` 组下可用的选项。

表 8.16. ml2_type_flat

配置选项 = 默认值	类型	Description
<code>flat_networks = *</code>	列表值	创建扁平网络的 <code>physical_network</code> 名称列表。使用 <code>default *</code> 允许带有任意 <code>physical_network</code> 名称的扁平网络。使用空列表禁用扁平网络。

8.6.4. ml2_type_geneve

下表概述了 `/etc/neutron/plugins/ml2/ml2_conf.ini` 文件中的 `[ml2_type_geneve]` 组下可用的选项。

表 8.17. ml2_type_geneve

配置选项 = 默认值	类型	Description
<code>max_header_size = 30</code>	整数值	Geneve 封装标头大小为动态的，这个值用于计算驱动程序的最大 MTU。此字段的默认大小为 30，这是 Geneve 标头的大小，而无需额外的选项标头。
<code>vni_ranges = []</code>	列表值	以逗号分隔的 <code><vni_min>:<vni_max></code> tuples 枚举 Geneve VNI ID 范围列表，它们可用于租户网络分配

8.6.5. ml2_type_gre

下表概述了 `/etc/neutron/plugins/ml2/ml2_conf.ini` 文件中的 `[ml2_type_gre]` 组下可用的选项。

表 8.18. ml2_type_gre

配置选项 = 默认值	类型	Description
<code>tunnel_id_ranges = []</code>	列表值	以逗号分隔的 <code><tun_min>:<tun_max></code> tuples 枚举可用于租户网络分配的 GRE 隧道 ID 范围

8.6.6. ml2_type_vlan

下表概述了 `/etc/neutron/plugins/ml2/ml2_conf.ini` 文件中的 `[ml2_type_vlan]` 组下可用的选项。

表 8.19. ml2_type_vlan

配置选项 = 默认值	类型	Description
<code>network_vlan_ranges = []</code>	列表值	<code><physical_network>:<vlan_min>:<vlan_max></code> 或 <code><physical_network></code> 指定可用于 VLAN 提供程序和租户网络的 <code>physical_network</code> 名称，以及每个可用于分配给租户网络的 VLAN 标签范围。

8.6.7. ml2_type_vxlan

下表概述了 `/etc/neutron/plugins/ml2/ml2_conf.ini` 文件中的 `[ml2_type_vxlan]` 组下可用的选项。

表 8.20. ml2_type_vxlan

配置选项 = 默认值	类型	Description
<code>vni_ranges = []</code>	列表值	以逗号分隔的 <code><vni_min>:<vni_max></code> tuples 枚举的 VXLAN VNI ID 范围列表，它们可用于租户网络分配
<code>vxlan_group = None</code>	字符串值	VXLAN 的多播组。配置后，将启用将所有广播流量发送到此多播组。在未配置的情况下，将禁用多播 VXLAN 模式。

8.6.8. ovs_driver

下表概述了 `/etc/neutron/plugins/ml2/ml2_conf.ini` 文件中的 `[ovs_driver]` 组下可用的选项。

表 8.21. ovs_driver

配置选项 = 默认值	类型	描述
<code>vnic_type_prohibit_list = []</code>	列表值	以逗号分隔的 VNIC 类型列表，其支持由机制驱动程序进行管理。请注意，支持的 <code>vnic_types</code> 取决于您的网络接口卡、您的操作系统的内核版本，以及 OVS 版本等其他因素。如果是 ovs 机制驱动程序，有效的 <code>vnic</code> 类型是 <code>normal</code> 和 <code>direct</code> 。请注意，直接只支持内核 4.8 和 ovs 2.8.0。绑定 DIRECT (SR-IOV) 端口允许使用 <code>tc</code> 将 OVS 流卸载到 SR-IOV NIC。这允许通过 <code>tc</code> 支持硬件卸载，并允许我们使用 <code>representor net-device</code> 通过 OpenFlow control plane 管理 VF。

8.6.9. securitygroup

下表概述了 `/etc/neutron/plugins/ml2/ml2_conf.ini` 文件中的 `[securitygroup]` 组下可用的选项。

表 8.22. securitygroup

配置选项 = 默认值	类型	Description
<code>enable_ipset = True</code>	布尔值	使用 <code>ipset</code> 来加快基于 <code>iptables</code> 的安全组。启用 <code>ipset</code> 支持要求在 L2 代理节点上安装 <code>ipset</code> 。
<code>enable_security_group = True</code>	布尔值	控制服务器中是否启用了 <code>neutron</code> 安全组 API。在不使用安全组或使用 <code>nova</code> 安全组 API 时，它应当为 <code>false</code> 。
<code>firewall_driver = None</code>	字符串值	L2 代理中的安全组防火墙的驱动程序

配置选项 = 默认值	类型	Description
permitted_ethertypes = []	列表值	允许以逗号分隔的 ethertypes 列表，以十六进制表示十六进制（从 "0x" 开始）。例如，"0x4008" 以允许 InfiniBand。

8.6.10. sriov_driver

下表概述了 `/etc/neutron/plugins/ml2/ml2_conf.ini` 文件中的 `[sriov_driver]` 组下可用的选项。

表 8.23. sriov_driver

配置选项 = 默认值	类型	描述
vnictype_prohibit_list = []	列表值	以逗号分隔的 VNIC 类型列表，其支持由机制驱动程序进行管理。请注意，支持的 vnic_types 取决于您的网络接口卡、您的操作系统的内核版本，以及其它因素。如果 sriov 机制驱动，有效的 VNIC 类型是 direct, macvtap 和 direct-physical。

8.7. NEUTRON.CONF

本节包含 `/etc/neutron/neutron.conf` 文件的选项。

8.7.1. DEFAULT

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[DEFAULT]` 组下可用的选项。

配置选项 = 默认值	类型	Description
agent_down_time = 75	整数值	有关代理关闭的秒数；应该至少为 report_interval 两次，以确保代理正常关闭。
allow_automatic_dhcp_failover = True	布尔值	从离线 DHCP 代理自动删除网络。
allow_automatic_l3agent_failover = False	布尔值	自动将路由器从离线 L3 代理重新调度到在线 L3 代理。
allow_bulk = True	布尔值	允许使用批量 API
allow_overlapping_ips = False	布尔值	在 Neutron 中允许重叠的 IP 支持。注意：如果 Neutron 与 Nova 安全组结合使用，则以下参数必须设为 False。

配置选项 = 默认值	类型	Description
allowed_contrack_helpers = <code>[{'amanda': 'tcp'}, {'ftp': 'tcp'}, {'h323': 'udp'}, {'h323': 'tcp'}, {'irc': 'tcp'}, {'netbios-ns': 'udp'}, {'pptp': 'tcp'}, {'sane': 'tcp'}, {'sip': 'udp'}, {'sip': 'tcp'}, {'snmp': 'udp'}, {'tftp': 'udp'}]</code>	列表值	定义允许的 contrack 帮助程序和 contrack helper 模块协议约束。
<code>api_extensions_path =</code>	字符串值	API 扩展的路径。请注意，这可以是以冒号分隔的路径列表。例如： <code>api_extensions_path = extensions:/path/to/more/exts:/even/more/exts</code> 。 <code>neutron.extensions</code> 的路径会附加到此后面，因此如果您的扩展位于那里，则不需要在此处指定它们。
api_paste_config = <code>api-paste.ini</code>	字符串值	api 服务的 <code>paste.deploy</code> 配置的文件名
api_workers = <code>None</code>	整数值	服务的独立 API worker 进程数量。如果未指定，则默认为等于可用 CPU 数量以获得最佳性能，按潜在的 RAM 使用量上限。
auth_strategy = <code>keystone</code>	字符串值	要使用的身份验证类型
backdoor_port = <code>None</code>	字符串值	启用事件程序后端。可接受的值为 <code>0</code> 、 <code><port></code> 和 <code><start>:<end></code> ，其中 <code>0</code> 会导致侦听随机的 tcp 端口号； <code><port></code> 会导致监听指定的端口号（如果正在使用该端口，则不会启用 backdoor）； <code><start>:<end></code> 会在指定的端口号范围内侦听最小未使用的端口号。所选端口显示在服务的日志文件中。
backdoor_socket = <code>None</code>	字符串值	启用 eventlet 后端，使用提供的路径作为可接收连接的 unix 套接字。这个选项与 <code>backdoor_port</code> 相互排斥，该选项只应提供一个。如果提供这两个选项，则此选项的存在会覆盖该选项的使用。在路径 <code>{pid}</code> 中将被当前进程的 PID 替换。
backlog = <code>4096</code>	整数值	配置套接字的积压请求数
base_mac = <code>fa:16:3e:00:00:00</code>	字符串值	Neutron 将用于 VIF 的基本 MAC 地址。前 3 个八位字节将保持不变。如果第 4 个八位字节不是 <code>00</code> ，它将被使用。其他对象将随机生成。
bind_host = <code>0.0.0.0</code>	主机地址值	要绑定到的主机 IP。
bind_port = <code>9696</code>	端口值	要绑定到的端口

配置选项 = 默认值	类型	Description
client_socket_timeout = 900	整数值	客户端连接的套接字操作的超时。如果传入的连接闲置了这个秒数，它将被关闭。值 0 表示等待永久。
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
control_exchange = neutron	字符串值	默认交换，哪些主题在哪些方面有范围。可以被 transport_url 选项指定的交换名称覆盖。
core_plugin = None	字符串值	Neutron 将使用的核心插件
debug = False	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。
default_availability_zones = []	列表值	可用区提示的默认值。当资源 availability_zone_hints 为空时，可用性区域了解调度程序会使用它。可以使用逗号分隔的字符串来指定多个可用区。这个值可以为空。在这种情况下，即使资源的 availability_zone_hints 为空，在调度资源时将可用性区域视为高可用性。

配置选项 = 默认值	类型	Description
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpidd=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 log_config_append，则忽略此选项。
dhcp_agent_notification = True	布尔值	允许向 DHCP 代理发送资源操作通知
dhcp_agents_per_networ k = 1	整数值	调度用于托管租户网络的 DHCP 代理数量。如果这个数字大于 1，调度程序会自动为给定租户网络分配多个 DHCP 代理，为 DHCP 服务提供高可用性。但是，这并不为隔离网络中的 IPv6 元数据服务提供高可用性。
dhcp_lease_duration = 86400	整数值	DHCP 租期持续时间（以秒为单位）。使用 -1 告知 dnsmasq 使用无限租期时间。

配置选项 = 默认值	类型	Description
dhcp_load_type = networks	字符串值	代表代理正在报告其负载的资源类型。这可以是 "networks", "subnets" 或 "ports"。指定（默认是网络），服务器将从代理报告状态中提取作为代理配置对象的一部分发送的特定负载，即 network_scheduler_driver 为 WeightScheduler 时，每个 report_interval.dhcp_load_type 都可以与 network_scheduler_driver = neutron.scheduler.dhcp_agent_scheduler.WeightScheduler when network_scheduler_driver 为 WeightScheduler, dhcp_load_type 结合使用，以表示资源的选择。示例：dhcp_load_type=networks
dns_domain = openstacklocal	字符串值	用于构建主机名的域
dvr_base_mac = fa:16:3f:00:00:00	字符串值	Neutron 用于唯一 DVR 实例的基础 mac 地址。前 3 个八位字节将保持不变。如果第 4 个八位字节不是 00，它将被使用。其他对象将随机生成。dvr_base_mac 必须与 base_mac 不同，以避免将它们与分配给租户端口的 MAC 混合。一个 4 个 octet 示例将是 dvr_base_mac = fa:16:3f:4f:00:00。默认值为 3 octet
enable_dvr = True	布尔值	确定是否为 DVR 配置了设置。如果为 False，则禁用 DVR API 扩展。
enable_new_agents = True	布尔值	当 enable_new_agents=False 时，代理以 admin_state_up=False 开头。在这种情况下，在 admin 将 admin_state_up 改为 True 前，用户的资源不会自动调度到代理。
enable_services_on_agents_with_admin_state_down = False	布尔值	在带有 admin_state_up False 的代理上启用服务。如果此选项为 False，当代理的 admin_state_up 变为 False 时，它将禁用它的服务。无论此选项是什么，不会选择具有 admin_state_up False 的代理进行自动调度。但是，如果此选项为 True，则手动调度到这些代理可用。
enable_snat_by_default = True	布尔值	如果 external_gateway_info 中未提供，定义 enable_snat 的默认值。
enable_traditional_dhcp = True	布尔值	如果为 False，neutron-server 将禁用以下 DHCP-agent 相关功能：1. DHCP 置备块 2.DHCP 调度程序 API 扩展 3. 网络调度机制 4.DHCP RPC/notification
executor_thread_pool_size = 64	整数值	当 executor 为 threading 或 eventlet 时，executor 线程池的大小。

配置选项 = 默认值	类型	Description
external_dns_driver = None	字符串值	用于外部 DNS 集成的驱动。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
filter_validation = True	布尔值	如果为 True，则允许插件决定是否对过滤器参数执行验证。如果打开此配置且所有插件都支持，则启用过滤验证
global_physnet_mtu = 1500	整数值	底层物理网络的 MTU。Neutron 使用这个值为所有虚拟网络组件计算 MTU。对于扁平化和 VLAN 网络，neutron 在不修改的情况下使用这个值。对于 VXLAN 等覆盖网络，neutron 会自动从这个值中减去覆盖协议开销。默认为 1500，以太网的标准值。
graceful_shutdown_timeout = 60	整数值	指定安全关闭服务器将退出的超时。零值表示无限期等待。
Host = <based on operating system>	主机地址值	此计算机上运行的 Neutron 服务器、代理和服务使用的主机名。在此计算机上运行的所有代理和服务都必须使用相同的主机值。
host_dvr_for_dhcp = True	布尔值	标志，以确定是否需要将 DVR 本地路由器托管到 DHCP 代理。如果 False，则 DHCP 代理实例支持的任何 L3 功能都不可能，例如：DNS。
http_retries = 3	整数值	在失败的 HTTP 调用中，应重试客户端连接(nova、ironic)的次数。0（零）表示连接只尝试一次（不重试）。设置为任何正整数表示连接失败时会多次重试。例如，将 设置为 3 表示连接总数为 4。
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。
<code>`instance_uuid_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。
interface_driver = None	字符串值	用于管理虚拟接口的驱动程序。
ipam_driver = internal	字符串值	要使用的 Neutron IPAM (IP 地址管理)驱动程序。默认情况下，使用 Neutron IPAM 驱动程序的引用实现。

配置选项 = 默认值	类型	Description
ipv6_pd_enabled = False	布尔值	为自动子网 CIDR 分配启用 IPv6 Prefix Delegation。设置为 True，在 PD 兼容环境中为子网分配启用 IPv6 前缀委派。在不提供 CIDR 或 subnetpool ID 的情况下为 IPv6 子网发出子网创建请求的用户将通过前缀委派机制授予 CIDR。请注意，启用 PD 将覆盖默认 IPv6 subnetpool 的行为。
l3_ha = False	布尔值	为虚拟路由器启用 HA 模式。
l3_ha_net_cidr = 169.254.192.0/18	字符串值	用于 l3 HA 管理网络的子网。
<code>l3_ha_network_physical_name = `</code>	字符串值	可以创建 HA 网络的物理网络名称。
<code>l3_ha_network_type = `</code>	字符串值	为 HA 路由器创建 HA 网络时要使用的网络类型。默认情况下，或为空，则使用第一个 <i>tenant_network_types</i> 。当 VRRP 流量应使用不是默认网络的特定网络时，这非常有用。
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 <i>log-date-format</i> ）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <i>%(asctime)s</i> 的格式字符串。默认： <i>%(default)s</i> 。如果设置了 <i>log_config_append</i> ，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 <i>log_file</i> 路径的基础目录。如果设置了 <i>log_config_append</i> ，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <i>use_stderr</i> 定义的 <i>stderr</i> 。如果设置了 <i>log_config_append</i> ，则忽略此选项。
log_options = True	布尔值	启动服务时，启用或禁用所有注册选项的日志值（在 DEBUG 级别）。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <i>log_rotation_type</i> 设置为 "interval"，否则此选项将被忽略。

配置选项 = 默认值	类型	Description
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 DEBUG 时，要附加到日志消息的其他数据。 oslo_log.formatters.ContextFormatter 使用
logging_default_format_s tring = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_for mat = %(user) s %(tenant) s %(domain) s % (user_domain) s % (project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_allowed_address_pa ir = 10	整数值	允许的地址对的最大数量
max_dns_nameservers = 5	整数值	每个子网的 DNS 名称服务器的最大数量
max_header_line = 16384	整数值	要接受的最大消息标头大小。在使用大令牌时可能需要增加 max_header_line（通常是在 keystone 配置为使用具有大型服务目录的 PKI 令牌时生成的）。
max_l3_agents_per_rout er = 3	整数值	在其上调度 HA 路由器的最大 L3 代理数。如果设为 0，则路由器将调度到每个代理上。

配置选项 = 默认值	类型	Description
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
max_routes = 30	整数值	每个路由器的最大路由数
max_subnet_host_routes = 20	整数值	每个子网的最大主机路由数
<code>`metadata_proxy_group = `</code>	字符串值	在初始化后运行元数据代理的组(gid 或名称)（如果为空：代理有效组）。
metadata_proxy_socket = \$state_path/metadata_proxy	字符串值	元数据代理 UNIX 域套接字的位置。
<code>`metadata_proxy_user = `</code>	字符串值	用户(uid 或名称)在其初始化后运行元数据代理（如果为空：代理有效的用户）。
migration_mode = False	布尔值	选项告知环境正在进行从 OVS 迁移到 OVN 的机制驱动程序。
network_auto_schedule = True	布尔值	允许自动调度网络到 DHCP 代理。
network_link_prefix = None	字符串值	此字符串前面带有到 OpenStack Network API 链接返回的普通 URL。如果为空（默认），则 URL 会以保持不变。
network_scheduler_driver = neutron.scheduler.dhcp_agent_scheduler.WeightScheduler	字符串值	用于将网络调度到 DHCP 代理的驱动程序
notify_nova_on_port_data_changes = True	布尔值	当端口数据(fixed_ips/floatingip)更改时向 nova 发送通知，以便 nova 可以更新其缓存。
notify_nova_on_port_status_changes = True	布尔值	当端口状态发生变化时向 nova 发送通知
pagination_max_limit = -1	字符串值	单个响应中返回的最大项目数，值为 <i>无限</i> 或负整数表示没有限制

配置选项 = 默认值	类型	Description
periodic_fuzzy_delay = 5	整数值	启动定期任务调度程序时随机延迟的秒数，以减少时间戳。（通过设置为 0 来禁用）
periodic_interval = 40	整数值	运行定期任务之间的秒。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
retry_until_window = 30	整数值	重试要侦听的秒数
router_auto_schedule = True	布尔值	允许自动将路由器调度到 L3 代理。
router_distributed = False	布尔值	系统范围的标志，以确定租户可以创建的路由器类型。只有管理员才能覆盖。
router_scheduler_driver = neutron.scheduler.l3_agent_scheduler.LeastRoutersScheduler	字符串值	用于将路由器调度到默认 L3 代理的驱动程序
rpc_conn_pool_size = 30	整数值	RPC 连接池的大小。
rpc_ping_enabled = False	布尔值	添加端点以回答 ping 调用。端点名为 oslo_rpc_server_ping
rpc_response_max_timeout = 600	整数值	从 RPC 调用等待响应的最大秒数。
rpc_response_timeout = 60	整数值	从调用等待响应的秒数。
rpc_state_report_workers = 1	整数值	专用于状态报告队列的 RPC worker 进程数量。
rpc_workers = None	整数值	服务的 RPC worker 进程数。如果没有指定，则默认等于 API worker 的数量。

配置选项 = 默认值	类型	Description
run_external_periodic_tasks = True	布尔值	某些定期任务可以在单独的进程中运行。我们是否应该在此处运行？
send_events_interval = 2	整数值	如果有要发送的事件，则发送事件到 nova 之间的秒数。
service_plugins = []	列表值	Neutron 将使用的服务插件
setproctitle = on	字符串值	将进程名称设置为与子 worker 角色匹配。可用选项包括： <i>off</i> - 保留之前的行为； <i>在 - 上</i> ，将进程重命名为 <i>neutron-server: role</i> （原始字符串）； <i>brief</i> - 在 <i>上</i> 重命名与 <i>上</i> 的相同，但没有原始字符串，如 <i>neutron-server: 角色</i> 。
state_path = /var/lib/neutron	字符串值	存储 Neutron 状态文件的位置。这个目录必须可以被代理写入。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 <code>log_config_append</code> ，则忽略此选项。
tcp_keepidle = 600	整数值	为每个服务器套接字设置 TCP_KEEPIIDLE 的值（以秒为单位）。在 OS X 上不支持。
transport_url = rabbit://	字符串值	用于连接到消息传递后端的网络地址和可选用户凭证，采用 URL 格式。预期格式为： driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query 示例： <code>rabbit://rabbitmq:password@127.0.0.1:5672//</code> 有关 URL 中字段的详情，请查看 <code>oslo_messaging.TransportURL</code> 文档 https://docs.openstack.org/oslo.messaging/latest/reference/transport.html
use-journal = False	布尔值	为日志记录启用 <code>journald</code> 。如果在 <code>systemd</code> 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 <code>log_config_append</code> ，此选项将被忽略。
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
use-syslog = False	布尔值	使用 <code>syslog</code> 进行日志记录。现有的 <code>syslog</code> 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 <code>log_config_append</code> ，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。

配置选项 = 默认值	类型	Description
use_ssl = False	布尔值	在 API 服务器上启用 SSL
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。
vlan_transparent = False	布尔值	如果为 True，则允许支持它的插件来创建 VLAN 透明网络。
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 log_file 选项并使用 Linux 平台时，才有意义。如果设置了 log_config_append，则忽略此选项。
wsgi_default_pool_size = 100	整数值	wsgi 使用的 greenthreads 池的大小
wsgi_keep_alive = True	布尔值	如果为 False，请明确关闭客户端套接字连接。
wsgi_log_format = %(client_ip) s "%(request_line) s" status: %(status_code) s len: %(body_length) s time: %(wall_seconds).7f	字符串值	用作模板用于生成日志行的 python 格式字符串。以下值可以格式化到其中：client_ip, date_time, request_line, status_code, body_length, wall_seconds。
wsgi_server_debug = False	布尔值	如果服务器应该将异常回溯发送到客户端 500 错误，则为 true。如果为 False，服务器将响应空正文。

8.7.2. agent

下表概述了 /etc/neutron/neutron.conf 文件中的 [agent] 组下可用的选项。

表 8.24. agent

配置选项 = 默认值	类型	Description
availability_zone = nova	字符串值	此节点的可用区
check_child_processes_action = respawn	字符串值	子进程结束时要执行的操作
check_child_processes_interval = 60	整数值	子进程存活度（秒）检查之间的间隔，使用 0 来禁用

配置选项 = 默认值	类型	Description
comment_iptables_rules = True	布尔值	为 iptables 规则添加注释。设置为 false 以禁止为生成的 iptables 规则添加注释，以描述每个规则的用途。系统必须支持 iptables 注释模块来添加注释。
debug_iptables_rules = False	布尔值	重复每个 iptables 差异计算，以确保生成的格式与 iptables-save 的格式匹配。对于生产环境的系统，不应该打开这个选项，因为它会给性能损失。
kill_scripts_path = /etc/neutron/kill_scripts/	字符串值	用于终止外部进程脚本的位置。此处的脚本名称必须遵循模式：“<process-name>-kill”，其中 <process-name> 是应该使用这个脚本被终止的进程名称。例如，dnsmasq 进程的终止脚本应命名为“dnsmasq-kill”。如果路径设定为 None，那么将使用默认“kill”命令来停止进程。
log_agent_heartbeats = False	布尔值	日志代理心跳
report_interval = 30	浮点值	节点报告状态之间的秒数；应该小于 agent_down_time，如果节点为一半或小于 agent_down_time。
root_helper = sudo	字符串值	根帮助程序应用。使用 <code>sudo neutron-rootwrap /etc/neutron/rootwrap.conf</code> 使用实际的 root 过滤器工具。更改为 <code>sudo</code> 以跳过过滤并仅直接运行命令。
root_helper_daemon = None	字符串值	尽可能要使用的根帮助程序守护进程应用程序。 使用 <code>sudo neutron-rootwrap-daemon /etc/neutron/rootwrap.conf</code> 以“daemon 模式”运行 rootwrap，该模式已报告以大规模提高性能。有关以“daemon 模式”运行 rootwrap 的更多信息，请参阅： https://docs.openstack.org/oslo.rootwrap/latest/user/usage.html#daemon-mode
use_helper_for_ns_read = True	布尔值	在列出系统上命名空间时，请使用 root 帮助程序。根据安全配置，可能不需要这样做。如果不需要 root 帮助程序，则将其设置为 False 以提高性能。
use_random_fully = True	布尔值	在 SNAT 伪装规则中使用随机。

8.7.3. CORS

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[cors]` 组下可用的选项。

表 8.25. CORS

配置选项 = 默认值	类型	Description
allow_credentials = True	布尔值	指明实际请求可以包含用户凭证
allow_headers = ['X-Auth-Token', 'X-Identity-Status', 'X-Roles', 'X-Service-Catalog', 'X-User-Id', 'X-Tenant-Id', 'X-OpenStack-Request-ID']	列表值	指定在实际请求过程中可以使用哪些标头字段名称。
allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH']	列表值	指明在实际请求期间可以使用哪些方法。
allowed_origin = None	列表值	指明此资源是否可以与请求"origin"标头中收到的域共享。格式："<protocol>://<host>[:<port>]", 没有尾部斜杠。示例： https://horizon.example.com
expose_headers = ['X-Auth-Token', 'X-Subject-Token', 'X-Service-Token', 'X-OpenStack-Request-ID', 'OpenStack-Volume-microversion']	列表值	指明哪些标头可以安全地公开给 API。默认为 HTTP Simple Headers。
max_age = 3600	整数值	CORS preflight 请求的最大缓存期限。

8.7.4. database

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[database]` 组下可用的选项。

表 8.26. database

配置选项 = 默认值	类型	Description
backend = sqlalchemy	字符串值	用于数据库的后端。
connection = None	字符串值	用于连接到数据库的 SQLAlchemy 连接字符串。
connection_debug = 0	整数值	SQL 调试信息的详细程度：0=None, 100=Everything。
<code>connection_parameters = `</code>	字符串值	用于连接时附加到连接 URL 的可选 URL 参数；指定 <code>param1=value1&param2=value2&...</code>
connection_recycle_time = 3600	整数值	连接池中存在的连接的时间超过这个秒数，下一次从池中签出它们的时间将替换为新的连接。

配置选项 = 默认值	类型	Description
connection_trace = False	布尔值	将 Python 堆栈跟踪作为注释字符串添加到 SQL。
db_inc_retry_interval = True	布尔值	如果为 True，将数据库操作的重试间隔增加到 db_max_retry_interval。
db_max_retries = 20	整数值	在引发错误前出现连接错误或死锁错误时的最大重试次数。设置为 -1 以指定无限重试计数。
db_max_retry_interval = 10	整数值	如果设置了 db_inc_retry_interval，则数据库操作重试之间的最大秒数。
db_retry_interval = 1	整数值	数据库事务重试之间的秒数。
<code>`engine = `</code>	字符串值	使用离线迁移时，将为其生成数据库引擎。
max_overflow = 50	整数值	如果设置，请将这个值用于 SQLAlchemy 的 max_overflow。
max_pool_size = 5	整数值	在池中保留的最大 SQL 连接数。将值设为 0 表示没有限制。
max_retries = 10	整数值	启动期间数据库连接重试的最大数量。设置为 -1 以指定无限重试计数。
mysql_enable_ndb = False	布尔值	如果为 True，则透明地启用对处理 MySQL 集群 (NDB) 的支持。
mysql_sql_mode = TRADITIONAL	字符串值	用于 MySQL 会话的 SQL 模式。这个选项（包括默认选项）会覆盖任何 server-set SQL 模式。要使用服务器配置设置的任何 SQL 模式，请将其设置为 no 值。 示例： <code>mysql_sql_mode=</code>
pool_timeout = None	整数值	如果设置，请将这个值用于 SQLAlchemy 的 pool_timeout。
retry_interval = 10	整数值	重新尝试打开 SQL 连接之间的间隔。
slave_connection = None	字符串值	用于连接到从数据库的 SQLAlchemy 连接字符串。
sqlite_synchronous = True	布尔值	如果为 True，则 SQLite 使用同步模式。
use_db_reconnect = False	布尔值	启用在连接丢失时进行数据库重新连接的实验性使用。

8.7.5. healthcheck

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[healthcheck]` 组下可用的选项。

表 8.27. healthcheck

配置选项 = 默认值	类型	Description
backends = []	列表值	可执行健康检查并报告该信息作为请求的一部分的其他后端。
detailed = False	布尔值	显示响应中更详细的信息。安全备注：启用此选项可能会公开有关被监控服务的敏感详情。务必验证它是否不会违反您的安全策略。
disable_by_file_path = None	字符串值	检查是否存在文件，以确定应用是否在端口上运行。DisableByFileHealthcheck 插件使用。
disable_by_file_paths = []	列表值	根据端口检查文件是否存在，以确定应用是否在端口上运行。需要字符串的 "port:path" 列表。DisableByFilesPortsHealthcheck 插件使用。
path = /healthcheck	字符串值	响应 healthcheck 请求的路径。

8.7.6. ironic

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[ironic]` 组下可用的选项。

表 8.28. ironic

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名

配置选项 = 默认值	类型	Description
enable_notifications = False	布尔值	将通知事件发送到 ironic。（例如，在相关端口状态上更改。）
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
split-loggers = False	布尔值	将请求记录到多个日志记录器。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名

8.7.7. keystone_authtoken

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[keystone_authtoken]` 组下可用的选项。

表 8.29. keystone_authtoken

配置选项 = 默认值	类型	Description
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
auth_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。 这个选项已弃用，现在使用 <code>www_authenticate_uri</code>，并将在 S 发行版本中删除。弃用自：Queens *reason:* <code>auth_uri</code> 选项已弃用，现在使用 <code>www_authenticate_uri</code>，并将在 S 发行版本中删除。
auth_version = None	字符串值	Identity API 端点的 API 版本。
cache = None	字符串值	存储 Swift 缓存对象的请求环境密钥。当使用 Swift 缓存部署 <code>auth_token</code> 中间件时，请使用此选项使中间件共享具有 <code>swift</code> 的缓存后端。否则，使用 <code>memcached_servers</code> 选项。
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。默认为系统 CA。
certFile = None	字符串值	如果身份服务器需要客户端证书，则需要此项
delay_auth_decision = False	布尔值	不要在中间件内处理授权请求，而是将授权决策委派给下游 WSGI 组件。
enforce_token_bind = permissive	字符串值	用于控制令牌绑定的使用和类型。可以设置为 "disabled" 以不检查令牌绑定。如果绑定类型是已知服务器的绑定类型，则验证绑定信息，并忽略它（如 "permissive"），但如果绑定类型未知，则令牌将被拒绝。"必需"令牌绑定需要允许任何形式的令牌绑定。最后，令牌中必须存在的绑定方法名称。
http_connect_timeout = None	整数值	请求超时值与身份 API 服务器通信。
http_request_max_retries = 3	整数值	与身份 API 服务器通信时，我们尝试重新连接的次数。
include_service_catalog = True	布尔值	（可选）指示是否设置 X-Service-Catalog 标头。如果为 False，则中间件不会要求在令牌验证时请求服务目录，且不会设置 X-Service-Catalog 标头。

配置选项 = 默认值	类型	Description
insecure = False	布尔值	验证 HTTPS 连接。
interface = internal	字符串值	用于 Identity API 端点的接口。有效值为 "public", "internal" (默认) 或 "admin"。
keyfile = None	字符串值	如果身份服务器需要客户端证书, 则需要此项
memcache_pool_conn_get_timeout = 10	整数值	(可选) 操作等待从池中获取 memcached 客户端连接的秒数。
memcache_pool_dead_retry = 300	整数值	(可选) 在再次尝试前 memcached 服务器被视为 dead 的秒数。
memcache_pool_maxsize = 10	整数值	(可选) 到每个 memcached 服务器的最大打开连接数。
memcache_pool_socket_timeout = 3	整数值	(可选) 与 memcached 服务器通信的套接字超时 (以秒为单位)。
memcache_pool_unused_timeout = 60	整数值	(可选) 与 memcached 的连接在池关闭前未使用的秒数。
memcache_secret_key = None	字符串值	(可选, 如果定义了 memcache_security_strategy) 此字符串用于密钥派生。
memcache_security_strategy = None	字符串值	(可选) 如果定义, 则指令牌数据是否应进行身份验证或加密。如果 MAC, 令牌数据在缓存中进行身份验证 (使用 HMAC)。如果 ENCRYPT, 令牌数据会在缓存中加密并进行身份验证。如果该值不是这些选项之一或为空, 则 auth_token 会在初始化时引发异常。
memcache_use_advanced_pool = False	布尔值	(可选) 使用 advanced (事件安全) memcached 客户端池。高级池只能在 python 2.x 下工作。
memcached_servers = None	列表值	(可选) 指定要用于缓存的 memcached 服务器列表。如果未定义, 令牌将会被缓存在进程中。
region_name = None	字符串值	找到身份服务器的区域。
service_token_roles = ['service']	列表值	需要出现在服务令牌中的角色。服务令牌可以请求可以使用过期的令牌, 因此此检查应严格控制只有实际服务应发送此令牌。此处的角色作为 ANY 检查应用, 因此此列表中的任何角色都必须存在。为了向后兼容的原因, 这目前只会影响 allow_expired 检查。

配置选项 = 默认值	类型	Description
service_token_roles_required = False	布尔值	为了向后兼容的原因，我们必须允许有效的服务令牌传递，但不允许将 service_token_roles 检查视为有效。在以后的发行版本中设置此 true 将成为默认设置，并应启用（如果可能）。
service_type = None	字符串值	服务的名称或类型（在服务目录中出现）。这用于验证具有受限访问规则的令牌。
token_cache_time = 300	整数值	为了防止过度努力验证令牌，中间件会缓存之前可预见的令牌（以秒为单位）。设置为 -1 可完全禁用缓存。
www_authenticate_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。

8.7.8. nova

下表概述了 `/etc/neutron/neutron.conf` 文件中的 **[nova]** 组下可用的选项。

表 8.30. nova

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID

配置选项 = 默认值	类型	Description
domain-name = None	字符串值	范围为的域名
endpoint_type = public	字符串值	要使用的 nova 端点的类型。此端点将在 keystone 目录中查找，并且应该是公共、内部或 admin 之一。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
live_migration_events = False	布尔值	启用此选项后，在实时迁移过程中，OVS 代理仅在目标主机接口绑定时才会发送 "vif-plugged-event"。这个选项还禁用在此置备端口时发送到 Nova 这个事件的任何其他代理（如 DHCP）。如果 Nova 补丁 https://review.opendev.org/c/openstack/nova/+767368 已就位，则可以启用此选项。此选项是临时的，并将在 Y 中删除，行为将是 "True"。
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region_name = None	字符串值	要使用的 nova 区域名称。如果 keystone 管理多个区域，则很有用。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID

配置选项 = 默认值	类型	Description
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名

8.7.9. oslo_concurrency

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[oslo_concurrency]` 组下可用的选项。

表 8.31. oslo_concurrency

配置选项 = 默认值	类型	Description
disable_process_locking = False	布尔值	启用或禁用进程间锁定。
lock_path = None	字符串值	用于锁定文件的目录。为安全起见，指定目录应只由运行需要锁定的进程的用户写入。默认为环境变量 <code>OSLO_LOCK_PATH</code> 。如果使用外部锁定，则必须设置锁定路径。

8.7.10. oslo_messaging_amqp

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[oslo_messaging_amqp]` 组下可用的选项。

表 8.32. oslo_messaging_amqp

配置选项 = 默认值	类型	Description
addressing_mode = dynamic	字符串值	表示驱动程序使用的寻址模式。允许的值： <code>legacy</code> - 使用旧的不可路由寻址； <code>routable</code> - 使用可路由地址； <code>dynamic</code> - 如果消息总线不支持路由，使用 <code>legacy</code> ，如果支持，使用 <code>routable</code> 。
anycast_address = anycast	字符串值	在发送到一组消费者时附加到地址前缀。消息总线用来识别应在消费者之间以轮循方式发送的消息。
broadcast_prefix = broadcast	字符串值	广播到所有服务器时使用的地址前缀
connection_retry_backoff = 2	整数值	每次失败故障转移尝试后，将 <code>connection_retry_interval</code> 增加到这秒数。
connection_retry_interval = 1	整数值	尝试重新连接前暂停的秒数。

配置选项 = 默认值	类型	Description
connection_retry_interval_max = 30	整数值	connection_retry_interval + connection_retry_backoff 的最大限制
container_name = None	字符串值	AMQP 容器的名称必须全局唯一。默认为生成的 UUID
default_notification_exchange = None	字符串值	通知地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_notification_exchange if set else control_exchange if set else <i>notify</i>
default_notify_timeout = 30	整数值	发送通知消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_reply_retry = 0	整数值	重新发送回复消息的最大尝试次数，因为可恢复错误而失败。
default_reply_timeout = 30	整数值	rpc 回复消息发送的截止时间。
default_rpc_exchange = None	字符串值	RPC 地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_rpc_exchange if set other control_exchange if set else <i>rpc</i>
default_send_timeout = 30	整数值	rpc cast 或调用消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_sender_link_timeout = 600	整数值	调度闲置发送者链接清除的持续时间。过期后分离链接。
group_request_prefix = unicast	字符串值	发送到组中任何服务器时的地址前缀
idle_timeout = 0	整数值	不活跃连接的超时时间（以秒为单位）
link_retry_delay = 10	整数值	重新连接 AMQP 1.0 链接之间的暂停时间，该链接因为可恢复的错误而失败。
multicast_address = multicast	字符串值	在发送异常消息时，附加到地址前缀。消息总线用来识别 fanout 消息。
notify_address_prefix = openstack.org/om/notify	字符串值	所有生成的通知地址的地址前缀
notify_server_credit = 100	整数值	传入通知消息的窗口大小

配置选项 = 默认值	类型	Description
pre_settled = ['rpc-cast', 'rpc-reply']	多值	发送此类型的消息预先组装。预先设置的消息不会接收来自 peer 的确认信息。请注意：如果发送失败，则预先设置的消息可能会被静默丢弃。允许的值： <i>rpc-call</i> - 发送 RPC Calls <i>pre-settled</i> <i>rpc-reply</i> - 发送 RPC Replies <i>pre-settled</i> <i>rpc-cast</i> - Send RPC Casts <i>pre-settled</i> <i>notify</i> - Send Notifications <i>pre-settled</i>
pseudo_vhost = True	布尔值	为不支持虚拟主机的消息总线启用虚拟主机支持（如 qpidd）。当设置为 true 时，虚拟主机名将添加到所有消息总线地址中，从而有效地为每个虚拟主机 <i>创建专用子网</i> 。如果消息总线支持使用 AMQP 1.0 Open performative 中的 <i>hostname</i> 字段作为虚拟主机的名称，设置为 False。
reply_link_credit = 200	整数值	传入 RPC 消息的窗口大小。
rpc_address_prefix = openstack.org/om/rpc	字符串值	所有生成的 RPC 地址的地址前缀
rpc_server_credit = 100	整数值	传入 RPC Request 消息的窗口大小
<code>`sasl_config_dir = `</code>	字符串值	包含 SASL 配置的目录路径
<code>`sasl_config_name = `</code>	字符串值	配置文件名称（不带 .conf 后缀）
<code>`sasl_default_realm = `</code>	字符串值	如果用户名中没有 realm，则要使用的 SASL 域
<code>`sasl_mechanisms = `</code>	字符串值	可接受的 SASL 机制的以空格分隔的列表
server_request_prefix = exclusive	字符串值	发送到特定服务器时使用的地址前缀
ssl = False	布尔值	尝试通过 SSL 连接。如果没有给出其他与 ssl 相关的参数，它将使用系统的 CA-bundle 来验证服务器证书。
<code>`ssl_ca_file = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_cert_file = `</code>	字符串值	自我标识用于客户端身份验证的证书 PEM 文件
<code>`ssl_key_file = `</code>	字符串值	用于为 <code>ssl_cert_file</code> 证书签名的私钥 PEM 文件（可选）
ssl_key_password = None	字符串值	用于解密 <code>ssl_key_file</code> 的密码（如果加密）

配置选项 = 默认值	类型	Description
ssl_verify_vhost = False	布尔值	默认情况下，SSL 检查服务器证书中的名称与 transport_url 中的主机名匹配。在某些配置中，最好使用虚拟主机名。例如，如果服务器使用 Server Name Indication TLS extension (rfc6066) 来为每个虚拟主机提供证书。如果服务器的 SSL 证书使用虚拟主机名称而不是 DNS 名称，则将 ssl_verify_vhost 设置为 True。
trace = False	布尔值	debug : 将 AMQP 帧转储到 stdout
unicast_address = unicast	字符串值	在发送到特定 RPC/Notification 服务器时，附加到地址前缀。消息总线用来识别发送到单个目的地的消息。

8.7.11. oslo_messaging_kafka

下表概述了 /etc/neutron/neutron.conf 文件中的 [oslo_messaging_kafka] 组下可用的选项。

表 8.33. oslo_messaging_kafka

配置选项 = 默认值	类型	Description
compression_codec = none	字符串值	producer 生成的所有数据的压缩 codec。如果没有设置，则不会使用压缩。请注意，这允许的值取决于 kafka 版本
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
consumer_group = oslo_messaging_consumer	字符串值	Kafka 消费者的组 ID。一个组中的消费者将协调消息消耗
enable_auto_commit = False	布尔值	启用异步消费者提交
kafka_consumer_timeout = 1.0	浮点值	Kafka 用户的默认超时
kafka_max_fetch_bytes = 1048576	整数值	Kafka consumer 的最大获取字节数
max_poll_records = 500	整数值	轮询调用中返回的最大记录数
pool_size = 10	整数值	Kafka Consumers 的池大小

配置选项 = 默认值	类型	Description
producer_batch_size = 16384	整数值	producer async 发送的批处理大小
producer_batch_timeout = 0.0	浮点值	以秒为单位的 KafkaProducer 批处理延迟上限
sasl_mechanism = PLAIN	字符串值	当安全协议是 SASL 时的机制
security_protocol = PLAINTEXT	字符串值	用于与代理通信的协议
<code>`ssl_cafile = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_client_cert_file = `</code>	字符串值	用于身份验证的客户端证书 PEM 文件。
<code>`ssl_client_key_file = `</code>	字符串值	用于身份验证的客户端密钥 PEM 文件。
<code>`ssl_client_key_password = `</code>	字符串值	用于身份验证的客户端密钥密码文件。

8.7.12. oslo_messaging_notifications

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[oslo_messaging_notifications]` 组下可用的选项。

表 8.34. oslo_messaging_notifications

配置选项 = 默认值	类型	Description
driver = []	多值	处理发送通知的驱动程序。可能的值有 messaging, messagingv2, routing, log, test, noop
retry = -1	整数值	重新发送因为可恢复错误而发送失败的通知消息的最大尝试次数。0 - 不重试, -1 - indefinite
topics = ['notifications']	列表值	用于 OpenStack 通知的 AMQP 主题。
transport_url = None	字符串值	代表用于通知的消息传递驱动程序的 URL。如果没有设置, 我们回退到用于 RPC 的同一配置。

8.7.13. oslo_messaging_rabbit

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[oslo_messaging_rabbit]` 组下可用的选项。

表 8.35. oslo_messaging_rabbit

配置选项 = 默认值	类型	Description
amqp_auto_delete = False	布尔值	AMQP 中的自动删除队列。
amqp_durable_queues = False	布尔值	在 AMQP 中使用持久队列。
direct_mandatory_flag = True	布尔值	(DEPRECATED)为直接发送启用/禁用 RabbitMQ 强制标记。直接发送用作回复，因此在客户端队列不存在时引发 MessageUndeliverable 异常。MessageUndeliverable 异常将用于循环使发送者恢复。此标志已弃用，且无法停用此功能
enable_cancel_on_failover = False	布尔值	启用 x-cancel-on-ha-failover 标志，使 rabbitmq 服务器将取消并通知 consumerswhen 队列停机
heartbeat_in_pthread = False	布尔值	默认情况下，通过原生 python 线程运行健康检查心跳线程。如果此选项等于 False，则健康检查 heartbeat 将继承父进程的执行模型。例如，如果父进程使用 eventlet/greenlet 对 stdlib 进行补丁，则心跳将通过绿色线程运行。这个选项应该只针对 wsgi 服务设置为 True。
heartbeat_rate = 2	整数值	heartbeat_timeout_threshold 期间的我们检查心跳的频率。
heartbeat_timeout_threshold = 60	整数值	如果 heartbeat 的 keep-alive 失败(0 禁用 heartbeat)，则 Rabbit 代理被视为 down 的秒数。
kombu_compression = None	字符串值	EXPERIMENTAL : Possible 值为 : gzip、bz2。如果没有设置压缩，则不会使用。这个选项可能在以后的版本中不可用。
kombu_failover_strategy = round-robin	字符串值	确定当当前连接到的 RabbitMQ 节点不可用时如何选择下一个 RabbitMQ 节点。仅在配置中提供多个 RabbitMQ 节点时生效。
kombu_missing_consumer_retry_timeout = 60	整数值	在放弃发送其回复前等待缺少的客户端的时间。这个值不应比 rpc_response_timeout 长。
kombu_reconnect_delay = 1.0	浮点值	响应 AMQP 消费者取消通知前等待多久。
rabbit_ha_queues = False	布尔值	尝试在 RabbitMQ 中使用 HA 队列(x-ha-policy: all)。如果更改这个选项，您必须擦除 RabbitMQ 数据库。在 RabbitMQ 3.0 中，在声明队列时，x-ha-policy 参数不再控制队列镜像。如果您只想确保所有队列（带有自动生成的名称的队列除外）都在所有节点上进行镜像，请运行： <code>"rabbitmqctl set_policy HA ^(?!amq\.)gem {"ha-mode": "all"}"</code>

配置选项 = 默认值	类型	Description
rabbit_interval_max = 30	整数值	RabbitMQ 连接重试的最大间隔。默认值为 30 秒。
rabbit_login_method = AMQPLAIN	字符串值	RabbitMQ 登录方法。
rabbit_qos_prefetch_count = 0	整数值	指定要预抓取的消息数量。设置为零可允许无限消息。
rabbit_retry_backoff = 2	整数值	连接到 RabbitMQ 时重试之间的 backoff 时长。
rabbit_retry_interval = 1	整数值	使用 RabbitMQ 重试连接的频率。
rabbit_transient_queues_ttl = 1800	整数值	正整数代表队列 TTL (x-expires) 的持续时间（以秒为单位）。TTL 期间未使用的队列将被自动删除。参数仅影响 reply 和 fanout 队列。
ssl = False	布尔值	通过 SSL 连接。
<code>`ssl_ca_file = `</code>	字符串值	SSL 证书颁发机构文件（仅在启用了 SSL 时评估）。
<code>`ssl_cert_file = `</code>	字符串值	SSL 证书文件（仅在启用了 SSL 时评估）。
<code>`ssl_key_file = `</code>	字符串值	SSL 密钥文件（仅在启用了 SSL 时 valid）。
<code>`ssl_version = `</code>	字符串值	要使用的 SSL 版本（仅在启用了 SSL 时评估）。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1_1 和 TLSv1_2 可能在某些发行版中提供。

8.7.14. oslo_middlewares

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[oslo_middlewares]` 组下可用的选项。

表 8.36. oslo_middlewares

配置选项 = 默认值	类型	Description
enable_proxy_headers_parsing = False	布尔值	应用程序是否在代理后面。这决定了中间件是否应该解析标头。

8.7.15. oslo_policy

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[oslo_policy]` 组下可用的选项。

表 8.37. oslo_policy

配置选项 = 默认值	类型	描述
enforce_new_defaults = False	布尔值	这个选项控制在评估策略时是否使用旧的弃用默认值。如果为 True ，则不会评估旧的已弃用的默认值。这意味着，如果旧默认值允许任何现有令牌，但不允许使用新默认值，它将被禁止。我们鼓励启用此标志以及 enforce_scope 标志，以便您可以同时获得新默认值和 scope_type 的好处
enforce_scope = False	布尔值	此选项控制在评估策略时是否强制执行范围。如果为 True ，则请求中使用的令牌范围将与被强制执行策略的 scope_types 进行比较。如果范围不匹配，则引发 InvalidScope 异常。如果为 False ，则会记录一条信息，告知操作员使用了不匹配的范围调用策略。
policy_default_rule = default	字符串值	默认规则。未找到请求的规则时强制执行。
policy_dirs = ['policy.d']	多值	存储策略配置文件的目录。它们可以相对于 <code>config_dir</code> 选项或绝对路径定义的搜索路径中的任何目录。由 <code>policy_file</code> 定义的文件必须存在，才能搜索这些目录。缺少或空目录将被忽略。
policy_file = policy.yaml	字符串值	将角色映射到给定服务权限的文件的相对或绝对路径。必须使用与配置文件设置相关的相对路径。
remote_content_type = application/x-www-form-urlencoded	字符串值	用于发送和接收基于 REST 策略检查的数据的内容类型
remote_ssl_ca_cert_file = None	字符串值	基于 REST 的策略检查的 ca cert 文件的绝对路径
remote_ssl_client_cert_file = None	字符串值	基于 REST 的策略检查的客户端证书的绝对路径
remote_ssl_client_key_file = None	字符串值	绝对路径客户端密钥文件 REST 基于策略检查
remote_ssl_verify_server_cert = False	布尔值	基于 REST 的策略检查的服务器身份验证

8.7.16. oslo_reports

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[oslo_reports]` 组下可用的选项。

表 8.38. oslo_reports

配置选项 = 默认值	类型	Description
file_event_handler = None	字符串值	要监视触发报告更改的文件路径，而不是信号。设置此选项会禁用报告的信号触发器。如果应用程序作为 WSGI 应用程序运行，则建议使用此选项而不是信号。
file_event_handler_interval = 1	整数值	设置 file_event_handler 时轮询之间等待的秒数
log_dir = None	字符串值	创建文件的路径

8.7.17. placement

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[placement]` 组下可用的选项。

表 8.39. placement

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
endpoint_type = public	字符串值	要使用的放置端点的类型。此端点将在 keystone 目录中查找，并且应该是公共、内部或 admin 之一。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件

配置选项 = 默认值	类型	Description
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region_name = None	字符串值	要使用的放置区域的名称。如果 keystone 管理多个区域，则很有用。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名

8.7.18. privsep

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[privsep]` 组下可用的选项。

表 8.40. privsep

配置选项 = 默认值	类型	Description
capabilities = []	列表值	privsep 守护进程保留的 Linux 功能列表。

配置选项 = 默认值	类型	Description
group = None	字符串值	privsep 守护进程应作为。
helper_command = None	字符串值	如果不使用"fork"方法，调用以启动 privsep 守护进程。如果没有指定，则使用"sudo privsep-helper"和用于重新创建当前配置的参数生成默认值。此命令必须接受合适的 --privsep_context 和 --privsep_sock_path 参数。
logger_name = oslo_privsep.daemon	字符串值	用于此 privsep 上下文的日志记录器名称。默认情况下，带有 oslo_privsep.daemon 的所有上下文日志。
thread_pool_size = <based on operating system>	整数值	可用于 privsep 的线程数量，以同时运行进程。默认为系统中的 CPU 内核数。
user = None	字符串值	privsep 守护进程应以。

8.7.19. profiler

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[profiler]` 组下可用的选项。

表 8.41. profiler

配置选项 = 默认值	类型	Description
connection_string = messaging://	字符串值	<p>通知程序后端的连接字符串。</p> <p>默认值为 messaging://，它将通知程序设置为 <code>oslo_messaging</code>。</p> <p>可能的值示例：</p> <ul style="list-style-type: none"> ● messaging:// - 使用 <code>oslo_messaging</code> 驱动程序来发送范围。 ● redis://127.0.0.1:6379 - 使用 <code>redis</code> 驱动程序发送 span。 ● MongoDB://127.0.0.1:27017 - 使用 <code>mongodb</code> 驱动程序进行发送范围。 ● Elasticsearch://127.0.0.1:9200 - 使用 <code>elasticsearch</code> 驱动程序来发送 span。 ● Jaeger://127.0.0.1:6831 - 使用 <code>jaeger tracing</code> 作为发送 span 的驱动程序。

配置选项 = 默认值	类型	Description
enabled = False	布尔值	<p>为此节点上的所有服务启用性能分析。</p> <p>默认值为 False（完全禁用性能分析功能）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用该功能 ● false：禁用该功能。分析无法通过此项目操作启动。如果性能分析由另一个项目触发，则此项目部分将为空。
es_doc_type = notification	字符串值	elasticsearch 中通知索引的文档类型。
es_scroll_size = 10000	整数值	Elasticsearch 批量分割大型请求。此参数定义每个批处理的最大大小（例如：es_scroll_size=10000）。
es_scroll_time = 2m	字符串值	此参数是一个时间值参数（例如：es_scroll_time=2m），指示参与搜索的节点将维护相关资源以继续和支持它。
filter_error_trace = False	布尔值	<p>启用包含 error/exception 的过滤器追踪来分隔的位置。</p> <p>默认值为 False。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用包含 error/exception 的过滤器追踪。 ● false：禁用过滤器。
hmac_keys = SECRET_KEY	字符串值	<p>用于加密上下文数据的 secret 密钥，以用于性能分析。</p> <p>此字符串值应该具有以下格式：<key1>[,<key2>,...<keyn>]，其中每个键都是随机字符串。通过 REST API 触发性能分析的用户必须在 REST API 调用的标头中设置其中一个键，使其包含此节点的性能分析结果。</p> <p>"enabled"标志和"hmac_keys"配置选项都应设置为启用性能分析。此外，为了跨所有服务生成正确的性能分析信息，OpenStack 项目之间至少需要一个关键。这样可确保它可用于客户端生成 trace，其中包含所有可能的资源的信息。</p>
sentinel_service_name = mymaster	字符串值	<p>Redis sentinel 使用服务名称来识别 master redis 服务。此参数定义名称（例如：sentinel_service_name=mymaster）。</p>

配置选项 = 默认值	类型	Description
socket_timeout = 0.1	浮点值	Redis Sentinel 在连接上提供了一个 timeout 选项。此参数定义该超时（例如：socket_timeout=0.1）。
trace_sqlalchemy = False	布尔值	<p>在服务中启用 SQL 请求性能分析。</p> <p>默认值为 False (SQL 请求不会被跟踪)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用 SQL 请求性能分析。每个 SQL 查询都将是 trace 的一部分，可以通过花费的时间来分析。 ● false：禁用 SQL 请求性能分析。花费的时间仅显示在更高级别的操作上。无法以这种方式分析单个 SQL 查询。

8.7.20. quotas

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[quotas]` 组下可用的选项。

表 8.42. quotas

配置选项 = 默认值	类型	Description
default_quota = -1	整数值	每个租户允许的默认资源数量。负值表示无限。
quota_driver = neutron.db.quota.driver_nolock.DbQuotaNoLockDriver	字符串值	用于配额检查的默认驱动程序。
quota_floatingip = 50	整数值	每个租户允许的浮动 IP 数量。负值表示无限。
quota_network = 100	整数值	每个租户允许的网络数。负值表示无限。
quota_port = 500	整数值	每个租户允许的端口数。负值表示无限。
quota_router = 10	整数值	每个租户允许的路由器数量。负值表示无限。
quota_security_group = 10	整数值	每个租户允许的安全组数。负值表示无限。

配置选项 = 默认值	类型	Description
quota_security_group_rule = 100	整数值	每个租户允许的安全规则数量。负值表示无限。
quota_subnet = 100	整数值	每个租户允许的子网数，负值表示无限。
track_quota_usage = True	布尔值	在当前资源配额使用情况的数据库中进行跟踪。不利用 neutron 数据库的插件应将此标志设置为 False。

8.7.21. ssl

下表概述了 `/etc/neutron/neutron.conf` 文件中的 `[ssl]` 组下可用的选项。

表 8.43. ssl

配置选项 = 默认值	类型	Description
ca_file = None	字符串值	用于验证连接客户端的 CA 证书文件。
cert_file = None	字符串值	安全启动服务器时要使用的证书文件。
密码 = None	字符串值	设置可用密码列表。值应该是 OpenSSL 密码列表格式的字符串。
key_file = None	字符串值	安全启动服务器时使用的私钥文件。
version = None	字符串值	要使用的 SSL 版本（仅在启用了 SSL 时评估）。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1_1 和 TLSv1_2 可能在某些发行版中提供。

8.8. OPENVSWITCH_AGENT.INI

本节包含 `/etc/neutron/plugins/ml2/openvswitch_agent.ini` 文件的选项。

8.8.1. DEFAULT

下表概述了 `/etc/neutron/plugins/ml2/openvswitch_agent.ini` 文件中的 `[DEFAULT]` 组下可用的选项。

配置选项 = 默认值	类型	Description
debug = False	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。

配置选项 = 默认值	类型	Description
default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpidd=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 log_config_append，则忽略此选项。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
`instance_format = [instance: %(uuid)s]`	字符串值	通过日志消息传递的实例格式。
`instance_uuid_format = [instance: %(uuid)s]`	字符串值	通过日志消息传递的实例 UUID 的格式。
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 log-date-format）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 %(asctime)s 的格式字符串。默认：%(default)s。如果设置了 log_config_append，则忽略此选项。

配置选项 = 默认值	类型	Description
log-dir = None	字符串值	(可选) 用于相对 log_file 路径的基础目录。如果设置了 log_config_append, 则忽略此选项。
log-file = None	字符串值	(可选) 要将日志输出发送到的日志文件名称。如果没有设置默认值, 日志记录将进入由 use_stderr 定义的 stderr。如果设置了 log_config_append, 则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 log_rotation_type 设置为 "interval", 否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时, 将使用上次文件更改的时间 (或服务启动的时间)。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [% (request_id) s % (user_identity) s] % (instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 DEBUG 时, 要附加到日志消息的其他数据。 oslo_log.formatters.ContextFormatter 使用
logging_default_format_s tring = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_for mat = %(user) s %(tenant) s %(domain) s % (user_domain) s % (project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_logfile_count = 30	整数值	轮转日志文件的最大数量。

配置选项 = 默认值	类型	Description
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
rpc_response_max_timeout = 600	整数值	从 RPC 调用等待响应的最大秒数。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 log_file 选项并使用 Linux 平台时，才有意义。如果设置了 log_config_append，则忽略此选项。

8.8.2. agent

下表概述了 /etc/neutron/plugins/ml2/openvswitch_agent.ini 文件中的 [agent] 组下可用的选项。

表 8.44. agent

配置选项 = 默认值	类型	Description
arp_responder = False	布尔值	支持本地 ARP 响应器。需要 OVS 2.1 和 ML2 l2population 驱动。允许交换机（在支持覆盖时）在本地响应 ARP 请求，而无需对覆盖执行昂贵的 ARP 广播。注意：如果 enable_distributed_routing 设为 True，则代理中的 arp_responder 将自动设置为 True，无论配置文件中的设置是什么。
baremetal_smartnic = False	布尔值	启用代理处理智能 NIC 端口。
dont_fragment = True	布尔值	在传输 GRE/VXLAN 隧道的传出 IP 数据包上设置或取消设置不分段(DF)位。
drop_flows_on_start = False	布尔值	在启动时重置流程表。把它设置为 True 将会导致流量中断。
enable_distributed_routing = False	布尔值	使 l2 代理在 DVR 模式下运行。
explicitly_egress_direct = False	布尔值	当设置为 True 时，接受的出口单播流量不使用操作 NORMAL。可接受的出口数据包将在最终的出口表中处理，直接输出单播流量的直接输出流。
extensions = []	列表值	要使用的扩展列表
l2_population = False	布尔值	使用 ML2 l2population 机制驱动程序了解远程 MAC 和 IP，并改进隧道可扩展性。
minimize_polling = True	布尔值	通过监控 ovsdb 以进行接口更改来最小化轮询。
ovsdb_monitor_respawn_interval = 30	整数值	在丢失了与它的通信后，在重新生成 ovsdb monitor 前等待的秒数。
tunnel_csum = False	布尔值	在传输 GRE/VXLAN 隧道的传出 IP 数据包上设置或取消设置隧道的隧道标头校验和。
tunnel_types = []	列表值	代理支持的网络类型(gre、vx 和/或 geneve)。
veth_mtu = 9000	整数值	veth 接口的 MTU 大小
vxlan_udp_port = 4789	端口值	用于 VXLAN 隧道的 UDP 端口。

8.8.3. network_log

下表概述了 `/etc/neutron/plugins/ml2/openvswitch_agent.ini` 文件中的 `[network_log]` 组下可用的选项。

表 8.45. network_log

配置选项 = 默认值	类型	Description
burst_limit = 25	整数值	每个 rate_limit 的最大数据包数量。
local_output_log_base = None	字符串值	输出代理侧的 logfile 路径，默认 syslog 文件。
rate_limit = 100	整数值	每秒数据包日志记录数上限。

8.8.4. OVS

下表概述了 `/etc/neutron/plugins/ml2/openvswitch_agent.ini` 文件中的 `[ovs]` 组下可用的选项。

表 8.46. OVS

配置选项 = 默认值	类型	Description
bridge_mappings = []	列表值	以逗号分隔的 <code><physical_network>:<bridge></code> tuples 将物理网络名称映射到代理的特定 Open vSwitch 网桥名称，以用于扁平化和 VLAN 网络。网桥名称的长度不应超过 11。每个网桥都必须存在，并且应将物理网络接口配置为端口。服务器上配置的所有物理网络应具有到每个代理上相应网桥的映射。注意：如果您从这个映射中删除网桥，请确保将其与集成网桥断开连接，因为它不再由代理管理。
datapath_type = system	字符串值	要使用的 OVS 数据路径。 <code>system</code> 是默认值，对应于内核数据路径。若要启用用户空间数据路径，请将此值设置为 <code>netdev</code> 。
disable_packet_marking = False	布尔值	禁用启用 QoS 扩展时标记数据包。将 OVS 与硬件卸载搭配使用时，需要启用此选项，直到 <code>skb_priority</code> 、 <code>skb_mark</code> 和输出队列字段受支持并可卸载。如果启用了这个选项，则 VirtIO 端口不会有速率 QoS 规则（带宽限制或最小带宽）。
int_peer_patch_port = patch-tun	字符串值	用于隧道网桥的集成网桥中的对等跳接端口。
integration_bridge = br-int	字符串值	要使用的集成网桥。不要更改此参数，除非有好理由：这是 OVS 集成网桥的名称。每个 hypervisor 都有一个。集成网桥充当虚拟补丁 bay。所有 VM VIFs 都附加到此网桥，然后根据其网络连接 修补。
local_ip = None	IP 地址值	本地覆盖（隧道）网络端点的 IP 地址。使用位于其中一个主机网络接口的 IPv4 或 IPv6 地址。这个值的 IP 版本必须与 neutron 服务器节点上的 ML2 插件配置文件中的 <code>overlay_ip_version</code> 选项值匹配。

配置选项 = 默认值	类型	Description
of_connect_timeout = 300	整数值	等待本地交换机连接控制器的超时时间（以秒为单位）。
of_inactivity_probe = 10	整数值	local 交换机连接到控制器的 inactivity_probe 间隔（以秒为单位）。0 代表禁用不活跃探测。
of_listen_address = 127.0.0.1	IP 地址值	用于侦听 OpenFlow 连接的地址。
of_listen_port = 6633	端口值	用于侦听 OpenFlow 连接的端口。
of_request_timeout = 300	整数值	等待单个 OpenFlow 请求的超时（以秒为单位）。
ovsdb_connection = tcp:127.0.0.1:6640	字符串值	OVSDB 后端的连接字符串。监控时，将用于所有 ovsdb 命令和 ovsdb-client
ovsdb_debug = False	布尔值	启用 OVSDB 调试日志
resource_provider_bandwidths = []	列表值	以逗号分隔的 <bridge><egress_bw><ingress_bw> tuples 列表，以给定方向显示给定网桥的可用带宽。方向从虚拟机角度来看。带宽以每秒 kilobits (kbps) 来衡量。网桥必须出现在 bridge_mappings 作为值。这里必须列出 bridge_mappings 中的所有网桥。对于此处未列出的网桥，我们都不会在放置中创建资源供应商，也不会报告清单。省略的方向意味着我们不会报告相应类的清单。
resource_provider_default_hypervisor = None	字符串值	用于查找资源提供程序父级的默认管理程序名称。如果没有设置这个选项，则使用规范名称
resource_provider_hypervisors = {}	dict 值	网桥映射到虚拟机监控程序：<bridge><hypervisor>, ... hypervisor 名称用于定位资源供应商树的父名称。只有在 hypervisor 名称与 resource_provider_default_hypervisor 配置选项值不同时，只需要在个别情况下设置，nova-compute 管理该 hypervisor。
resource_provider_inventory_defaults = {'allocation_ratio': 1.0, 'min_unit': 1, 'reserved': 0, 'step_size': 1}	dict 值	key:value 对，用于指定报告资源提供程序清单时使用的默认值。可能的键及其类型：allocation_ratio:float, max_unit:int, min_unit:int, reserved:int, step_size:int, 另请参阅: https://docs.openstack.org/api-ref/placement/#update-resource-provider-inventories 。
ssl_ca_cert_file = None	字符串值	与 OVSDB 交互时使用的证书颁发机构(CA)证书。使用前缀为 ovsdb_connection 的 "ssl:" 时需要

配置选项 = 默认值	类型	Description
ssl_cert_file = None	字符串值	与 OVSDB 交互时使用的 SSL 证书文件。使用前缀为 <code>ovsdb_connection</code> 的 "ssl:" 时需要
ssl_key_file = None	字符串值	与 OVSDB 交互时使用的 SSL 私钥文件。使用前缀为 <code>ovsdb_connection</code> 的 "ssl:" 时需要
tun_peer_patch_port = patch-int	字符串值	用于集成网桥的隧道网桥中的对等跳接端口。
tunnel_bridge = br-tun	字符串值	要使用的隧道网桥。
vhostuser_socket_dir = /var/run/openvswitch	字符串值	OVS vhost-user 套接字目录。

8.8.5. securitygroup

下表概述了 `/etc/neutron/plugins/ml2/openvswitch_agent.ini` 文件中的 **[securitygroup]** 组下可用的选项。

表 8.47. securitygroup

配置选项 = 默认值	类型	Description
enable_ipset = True	布尔值	使用 ipset 来加快基于 iptables 的安全组。启用 ipset 支持要求在 L2 代理节点上安装 ipset。
enable_security_group = True	布尔值	控制服务器中是否启用了 neutron 安全组 API。在不使用安全组或使用 nova 安全组 API 时，它应当为 false。
firewall_driver = None	字符串值	L2 代理中的安全组防火墙的驱动程序
permitted_ethertypes = []	列表值	允许以逗号分隔的 ethertypes 列表，以十六进制表示十六进制（从 "0x" 开始）。例如，"0x4008" 以允许 InfiniBand。

8.9. SRIOV_AGENT.INI

本节包含 `/etc/neutron/plugins/ml2/sriov_agent.ini` 文件的选项。

8.9.1. DEFAULT

下表概述了 `/etc/neutron/plugins/ml2/sriov_agent.ini` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
debug = False	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。
default_log_levels = ['amqp=WARN', 'amqp-lib=WARN', 'boto=WARN', 'qp-id=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O']	列表值	以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 log_config_append，则忽略此选项。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
`instance_format = [instance: %(uuid)s]`	字符串值	通过日志消息传递的实例格式。
`instance_uuid_format = [instance: %(uuid)s]`	字符串值	通过日志消息传递的实例 UUID 的格式。
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 log-date-format）。

配置选项 = 默认值	类型	Description
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	(可选) 用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	(可选) 要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [%(request_id)s %(user_identity)s] %(instance)s%(message)s	字符串值	使用上下文格式化用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_debug_format_sufffix = %(funcName)s %(pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 <code>DEBUG</code> 时，要附加到日志消息的其他数据。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_default_format_string = %(asctime)s.%(msecs)03d %(process)d %(levelname)s %(name)s [-] %(instance)s%(message)s	字符串值	上下文未定义时用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用
logging_exception_prefix = %(asctime)s.%(msecs)03d %(process)d ERROR %(name)s %(instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 <code>oslo_log.formatters.ContextFormatter</code> 使用

配置选项 = 默认值	类型	Description
logging_user_identity_format = %(user) s %(tenant) s %(domain) s %(user_domain) s %(project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
publish_errors = False	布尔值	启用或禁用错误事件的发布。
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
rpc_response_max_timeout = 600	整数值	从 RPC 调用等待响应的最大秒数。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
use-journal = False	布尔值	为日志记录启用 journald。如果在 systemd 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 log_config_append，此选项将被忽略。
use-json = False	布尔值	使用 JSON 格式进行日志记录。如果设置了 log_config_append，则忽略此选项。
use-syslog = False	布尔值	使用 syslog 进行日志记录。现有的 syslog 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 log_config_append，则忽略此选项。
use_eventlog = False	布尔值	日志输出到 Windows 事件日志。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。

配置选项 = 默认值	类型	Description
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 <code>log_file</code> 选项并使用 Linux 平台时，才有意义。如果设置了 <code>log_config_append</code> ，则忽略此选项。

8.9.2. agent

下表概述了 `/etc/neutron/plugins/ml2/sriov_agent.ini` 文件中的 `[agent]` 组下可用的选项。

表 8.48. agent

配置选项 = 默认值	类型	Description
extensions = []	列表值	要使用的扩展列表

8.9.3. sriov_nic

下表概述了 `/etc/neutron/plugins/ml2/sriov_agent.ini` 文件中的 `[sriov_nic]` 组下可用的选项。

表 8.49. sriov_nic

配置选项 = 默认值	类型	Description
exclude_devices = []	列表值	以逗号分隔的 <code><network_device>:<vfs_to_exclude></code> tuples 列表，将 <code>network_device</code> 映射到不应用于虚拟网络的、特定于节点的虚拟功能列表。 <code>vfs_to_exclude</code> 是用分号分隔的虚拟功能列表，以便从 <code>network_device</code> 中排除。映射中的 <code>network_device</code> 应该会显示在 <code>physical_device_mappings</code> 列表中。
physical_device_mappings = []	列表值	以逗号分隔的 <code><physical_network>:<network_device></code> tuples 将物理网络名称映射到用于 VLAN 网络的特定于代理的 SR-IOV 物理功能的物理网络设备接口。服务器上 <code>network_vlan_ranges</code> 中列出的所有物理网络应具有到每个代理上相应接口的映射。
resource_provider_bandwidths = []	列表值	以逗号分隔的 <code><network_device>:<egress_bw></code> tuples 列表，在给定方向中显示给定设备的可用带宽。方向从虚拟机角度来看。带宽以每秒 kilobits (kbps) 来衡量。该设备必须出现在 <code>physical_device_mappings</code> 作为值。这里不能列出 <code>physical_device_mappings</code> 中的所有设备。对于此处未列出的设备，我们不会在放置中创建资源供应商，也不会报告清单。省略的方向意味着我们不会报告相应类的清单。
resource_provider_default_hypervisor = None	字符串值	用于查找资源提供程序父级的默认管理程序名称。如果没有设置这个选项，则使用规范名称

配置选项 = 默认值	类型	Description
resource_provider_hypervisors = {}	dict 值	网络设备映射到 hypervisor: <network_device>: <hypervisor>,... hypervisor name 用于找到资源供应商树的父名称。只有在 hypervisor 名称与 resource_provider_default_hypervisor 配置选项值不同时，只需要在个别情况下设置，nova-compute 管理该 hypervisor。
resource_provider_inventory_defaults = {'allocation_ratio': 1.0, 'min_unit': 1, 'reserved': 0, 'step_size': 1}	dict 值	key:value 对，用于指定报告资源提供程序清单时使用的默认值。可能的键及其类型：allocation_ratio:float, max_unit:int, min_unit:int, reserved:int, step_size:int, 另请参阅: https://docs.openstack.org/api-ref/placement/#update-resource-provider-inventories 。

第 9 章 NOVA

以下章节包含有关 **nova** 服务中的配置选项的信息。

9.1. NOVA.CONF

本节包含 `/etc/nova/nova.conf` 文件的选项。

9.1.1. DEFAULT

下表概述了 `/etc/nova/nova.conf` 文件中的 **[DEFAULT]** 组下可用的选项。

配置选项 = 默认值	类型	Description
allow_resize_to_same_host = False	布尔值	允许目标机器与源进行调整大小。在单主机环境中测试时很有用。默认情况下，不允许其调整到同一主机上。将此选项设置为 true 会将相同的主机添加到目标选项中。如果允许 <code>ServerGroupAffinityFilter</code> 并需要调整大小，则也设置为 true。
arq_binding_timeout = 300	整数值	加速器请求(ARQ)绑定事件消息的超时时间。 等待 ARQ 绑定解析事件到达的秒数。事件表示实例的每个 ARQ 都绑定成功或无法绑定。如果没有到达，则实例启动会中止，但有一个例外。
backdoor_port = None	字符串值	启用事件程序后端。可接受的值为 0、 <code><port></code> 和 <code><start>:<end></code> ，其中 0 会导致侦听随机的 tcp 端口号； <code><port></code> 会导致监听指定的端口号（如果正在使用该端口，则不会启用 backdoor）； <code><start>:<end></code> 会在指定的端口号范围内侦听最小未使用的端口号。所选端口显示在服务的日志文件中。
backdoor_socket = None	字符串值	启用 eventlet 后端，使用提供的路径作为可接收连接的 unix 套接字。这个选项与 <code>backdoor_port</code> 相互排斥，该选项只应提供一个。如果提供这两个选项，则此选项的存在会覆盖该选项的使用。在路径 <code>{pid}</code> 中将被当前进程的 PID 替换。

配置选项 = 默认值	类型	Description
block_device_allocate_retries = 60	整数值	<p>在服务器创建过程中附加卷前检查卷的次数。</p> <p>当使用块设备映射创建服务器时，其中 source_type 是 空白、镜像或快照，target_type 是 卷，nova-compute 服务将创建一个卷，然后将其附加到服务器。在卷可以被附加前，它必须处于 "available"。这个选项控制在附加前检查创建的卷的次数。</p> <p>如果操作超时，如果块设备映射 delete_on_termination 值为 True，则卷将被删除。</p> <p>建议您在块存储服务中配置镜像缓存来加快此操作。 详情请查看 https://docs.openstack.org/cinder/latest/admin/blockstorage-image-volume-cache.html。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 60（默认） ● 如果值为 0，则进行一次尝试。 ● 对于任何值 > 0，总尝试为 (value + 1) <p>相关选项：</p> <ul style="list-style-type: none"> ● block_device_allocate_retries_interval - 控制检查之间的间隔
block_device_allocate_retries_interval = 3	整数值	<p>块设备分配在失败时重试的时间间隔（以秒为单位）。</p> <p>这个选项允许用户指定连续重试之间的时间间隔。block_device_allocate_retries 选项指定重试的最大数量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0：禁用选项。 ● 任何正整数（以秒为单位）。 <p>相关选项：</p> <ul style="list-style-type: none"> ● block_device_allocate_retries - 控制重试次数

配置选项 = 默认值	类型	Description
cert = self.pem	字符串值	<p>SSL 证书文件的路径。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● key ● ssl_only ● [console] ssl_ciphers ● [console] ssl_minimum_version
compute_driver = None	字符串值	<p>定义用于控制虚拟化的驱动程序。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● libvirt.LibvirtDriver ● fake.FakeDriver ● ironic.IronicDriver ● vmwareapi.VMwareVCDriver ● hyperv.HyperVDriver ● powervm.PowerVMDriver ● zvm.ZVMDriver
compute_monitors = []	列表值	<p>以逗号分隔的 monitor 列表，可用于获取计算指标。您可以使用 nova.compute.monitors mdadm 命名空间的 setuptools 入口点中的 alias/name。如果没有提供命名空间，则会假定使用 "cpu." 命名空间来向后兼容。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>每次只能载入每个命名空间的一个监控器（例如：cpu）。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 空列表将禁用该功能（默认）。 ● 启用 CPU 的示例值 <p>使用 virt 驱动程序变体的带宽监控</p> <pre>compute_monitors = cpu.virt_driver</pre>

配置选项 = 默认值	类型	Description
config_drive_format = iso9660	字符串值	<p>配置驱动器格式。</p> <p>配置驱动器格式，该格式将包含实例引导时附加到实例的元数据。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● 当出现以下备选方法之一时，这个选项是有意义的： <ol style="list-style-type: none"> 1. force_config_drive 选项设置为 true 2. 创建实例的 REST API 调用包含 config drive 选项的 enable 标志 3. 用于创建实例的镜像需要一个配置驱动器，这由该镜像的 img_config_drive 属性定义。 ● 运行 Hyper-V hypervisor 的计算节点可以配置为将配置驱动器作为 CD 驱动器附加。要将配置驱动器作为 CD 驱动器附加，请将 [hyperv] config_drive_cdrom 选项设置为 true。 <p>弃用自：19.0.0</p> <p>原因：此选项最初作为 libvirt 中 bug 的临时解决方案，即在 libvirt v1.2.17 中解析的错误。因此，这个选项不再需要或有用。</p>
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
console_host = <based on operating system>	字符串值	<p>用于连接到此主机上实例的控制台代理主机。它是控制台主机的公开可见名称。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 当前主机名（默认）或任何代表主机名的字符串。
control_exchange = nova	字符串值	默认交换，哪些主题在哪些方面有范围。可以被 transport_url 选项指定的交换名称覆盖。

配置选项 = 默认值	类型	Description
cpu_allocation_ratio = None	浮点值	<p>虚拟 CPU 和物理 CPU 分配比率。</p> <p>这个选项用于通过为 VCPU 清单配置分配比率，来影响放置 API 选择的主机。</p> <p>a. note::</p> <p>This option does not affect `PCPU` inventory, which cannot be overcommitted.</p> <p>b. note::</p> <p>If this option is set to something *other than* `None` or `0.0`, the allocation ratio will be overwritten by the value of this option, otherwise, the allocation ratio will not change. Once set to a non-default value, it is not possible to "unset" the config to get back to the default behavior. If you want to reset back to the initial value, explicitly specify it to the value of `initial_cpu_allocation_ratio`.</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何有效的正整数或浮点值 <p>相关选项：</p> <ul style="list-style-type: none"> initial_cpu_allocation_ratio
daemon = False	布尔值	作为后台进程运行。
debug = False	布尔值	如果设置为 true，则日志级别将设置为 DEBUG，而不是默认的 INFO 级别。
default_access_ip_network_name = None	字符串值	<p>用于为实例设置访问 IP 的网络名称。如果有多个 IP 可供选择，则会选择一个任意 IP。</p> <p>可能的值：</p> <ul style="list-style-type: none"> none（默认） 代表网络名称的任何字符串。

配置选项 = 默认值	类型	Description
default_availability_zone = nova	字符串值	<p>计算服务的默认可用区。</p> <p>此选项决定 <i>nova-compute</i> 服务的默认可用区，如果服务不属于与可用性区域元数据的聚合，则使用此选项。</p> <p>可能的值：</p> <ul style="list-style-type: none">● 代表现有可用区名称的任何字符串。
default_ephemeral_format = None	字符串值	<p>创建时将格式化 ephemeral_volume 的默认格式。</p> <p>可能的值：</p> <ul style="list-style-type: none">● ext2● ext3● ext4● xfs● NTFS（仅适用于 Windows 客户机）

配置选项 = 默认值	类型	Description
<pre>default_log_levels = ['amqp=WARN', 'amqpplib=WARN', 'boto=WARN', 'qpidd=WARN', 'sqlalchemy=WARN', 'suds=INFO', 'oslo.messaging=INFO', 'oslo_messaging=INFO', 'iso8601=WARN', 'requests.packages.urllib 3.connectionpool=WARN' , 'urllib3.connectionpool= WARN', 'websocket=WARN', 'requests.packages.urllib 3.util.retry=WARN', 'urllib3.util.retry=WARN', 'keystonemiddleware=WA RN', 'routes.middleware=WAR N', 'stevedore=WARN', 'taskflow=WARN', 'keystoneauth=WARN', 'oslo.cache=INFO', 'oslo_policy=INFO', 'dogpile.core.dogpile=INF O', 'glanceclient=WARN', 'oslo.privsep.daemon=IN FO']</pre>	列表值	<p>以 logger=LEVEL 的形式列出软件包日志级别列表。如果设置了 log_config_append，则忽略此选项。</p>
<pre>default_schedule_zone = None</pre>	字符串值	<p>实例的默认可用区。</p> <p>此选项决定了实例的默认可用区，当用户创建实例时没有指定实例时将使用它。在它们的生命周期内，实例将绑定到此可用区。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表现有可用区名称的任何字符串。 ● none，这意味着如果实例从一个计算节点移到另一个计算节点，则实例可以在其生命周期内从一个可用区移到另一个可用区。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [cinder]/cross_az_attach

配置选项 = 默认值	类型	Description
<p>disk_allocation_ratio = None</p>	<p>浮点值</p>	<p>虚拟磁盘与物理磁盘分配比率。</p> <p>这个选项用于通过为 DISK_GB 清单配置分配比率，来影响放置 API 选择的主机。</p> <p>配置后，如果比率大于 1.0，将会导致超额订阅可用物理磁盘。这在使用一个没有使用整个虚拟磁盘的镜像创建实例时更高效地打包实例（如稀疏或压缩镜像）。它可以设置为 0.0 和 1.0 之间的值，以便保留磁盘的百分比供实例使用。</p> <p>a. note::</p> <div style="border-left: 2px solid black; padding-left: 10px; margin-left: 20px;"> <p>If the value is set to <code>>1`</code>, we recommend keeping track of the free disk space, as the value approaching <code>0`</code> may result in the incorrect functioning of instances using it at the moment.</p> </div> <p>b. note::</p> <div style="border-left: 2px solid black; padding-left: 10px; margin-left: 20px;"> <p>If this option is set to something <i>*other than*</i> <code>`None`</code> or <code>0.0`</code>, the allocation ratio will be overwritten by the value of this option, otherwise, the allocation ratio will not change. Once set to a non-default value, it is not possible to "unset" the config to get back to the default behavior. If you want to reset back to the initial value, explicitly specify it to the value of <code>`initial_disk_allocation_ratio`</code>.</p> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何有效的正整数或浮点值 <p>相关选项：</p> <ul style="list-style-type: none"> ● initial_disk_allocation_ratio

配置选项 = 默认值	类型	Description
enable_new_services = True	布尔值	<p>在此主机上自动启用新的 nova-compute 服务。</p> <p>当新的 nova-compute 服务启动时，它将作为启用的服务在数据库中注册。有时，注册新计算服务处于 disabled 状态会很有用，然后在稍后的时间点启用它们。这个选项只为 nova-compute 服务设置此行为，它不会自动禁用 nova-conductor、nova-scheduler 或 nova-osapi_compute 等其他服务。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：注册自身后，每个新计算服务都会启用。 ● false：计算服务必须通过 os-services REST API 调用启用，或使用带有 nova service-enable <hostname> <binary> 的 CLI 来启用，否则它们未就绪。
enabled_apis = ['osapi_compute', 'metadata']	列表值	默认情况下要启用的 API 列表。
enabled_ssl_apis = []	列表值	<p>启用 SSL 的 API 列表。</p> <p>Nova 为 API 服务器提供 SSL 支持。enabled_ssl_apis 选项允许配置 SSL 支持。</p>
executor_thread_pool_size = 64	整数值	当 executor 为 threading 或 eventlet 时，executor 线程池的大小。
fatal_deprecations = False	布尔值	启用或禁用弃用的致命状态。
flat_injected = False	布尔值	这个选项决定网络设置信息是否在引导前注入虚拟机。虽然最初设计为仅由 nova-network 使用，但它也被 vmware virt 驱动程序用于控制网络信息是否注入虚拟机。当我们使用 config_drive 配置网络时，libvirt virt 驱动程序也会使用它，以控制网络信息是否注入虚拟机。

配置选项 = 默认值	类型	Description
force_config_drive = False	布尔值	<p>强制注入在配置驱动器上进行</p> <p>当此选项设置为 true 配置驱动器功能时，默认将强制启用，否则用户仍然可以通过 REST API 或镜像元数据属性启用配置驱动器。启动的实例不受这个选项的影响。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：强制使用配置驱动器，无论用户在 REST API 调用中的输入。 ● false：不要强制使用配置驱动器。配置驱动器仍可通过 REST API 或镜像元数据属性启用。 <p>相关选项：</p> <ul style="list-style-type: none"> ● 使用 <code>mkisofs_cmd</code> 标志设置安装 <code>genisoimage</code> 程序的路径。如果 <code>genisoimage</code> 与 <code>nova-compute</code> 服务位于同一个路径，则不需要设置此标志。 ● 要将配置驱动器与 Hyper-V 一起使用，您必须将 <code>mkisofs_cmd</code> 值设置为 <code>mkisofs.exe</code> 安装的完整路径。另外，您必须将 <code>hyperv</code> 配置部分中的 <code>qemu_img_cmd</code> 值设置为到 <code>qemu-img</code> 命令安装的完整路径。
force_raw_images = True	布尔值	<p>强制将后备镜像转换为 raw 格式。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：备份镜像文件将转换为原始镜像格式 ● false：不会转换后备镜像文件 <p>相关选项：</p> <ul style="list-style-type: none"> ● compute_driver：只有 <code>libvirt</code> 驱动程序使用这个选项。 ● [libvirt]/images_type：如果 <code>images_type</code> 是 <code>rbd</code>，则不允许将此选项设置为 <code>False</code>。详情请查看 https://bugs.launchpad.net/nova/+bug/1816686 错误。
graceful_shutdown_timeout = 60	整数值	<p>指定安全关闭服务器将退出的超时。零值表示无限期等待。</p>

配置选项 = 默认值	类型	Description
heal_instance_info_cache_interval = 60	整数值	<p>实例网络信息缓存更新之间的间隔。</p> <p>每个计算节点运行任务的时间（以秒为单位）查询其所有实例网络信息，然后使用该信息更新 Nova db。如果此选项设为 0，Nova 永远不会更新它的缓存。如果没有更新缓存，元数据服务和 nova-api 端点将代理与实例相关的网络数据。因此，不建议将这个选项设置为 0。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数（以秒为单位）。 任何值 mvapich0 将禁用同步。不建议这样做。
Host = <based on operating system>	字符串值	<p>此主机的主机名、FQDN 或 IP 地址。</p> <p>用作：</p> <ul style="list-style-type: none"> nova-compute worker 的 oslo.messaging 队列名称 我们将这个值用于发送到 neutron 的 binding_host。这意味着，如果您使用 neutron 代理，它应该具有相同的值。 Cinder 主机附加信息 <p>在 AMQP 密钥中必须有效。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 带有主机名、FQDN 或 IP 地址的字符串。默认为此主机的主机名。
initial_cpu_allocation_ratio = 16.0	浮点值	<p>初始虚拟 CPU 和物理 CPU 分配比率。</p> <p>这仅在初始为给定 nova-compute 服务创建 computes_nodes 表记录时使用。</p> <p>如需了解更多详细信息和用法场景，请参阅 https://docs.openstack.org/nova/latest/admin/configuration/schedulers.html。</p> <p>相关选项：</p> <ul style="list-style-type: none"> cpu_allocation_ratio

配置选项 = 默认值	类型	Description
initial_disk_allocation_ratio = 1.0	浮点值	<p>初始虚拟磁盘与物理磁盘分配比率。</p> <p>这仅在初始为给定 nova-compute 服务创建 computes_nodes 表记录时使用。</p> <p>如需了解更多详细信息和用法场景，请参阅 https://docs.openstack.org/nova/latest/admin/configuration/schedulers.html。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● disk_allocation_ratio
initial_ram_allocation_ratio = 1.5	浮点值	<p>初始虚拟 RAM 与物理 RAM 分配比率。</p> <p>这仅在初始为给定 nova-compute 服务创建 computes_nodes 表记录时使用。</p> <p>如需了解更多详细信息和用法场景，请参阅 https://docs.openstack.org/nova/latest/admin/configuration/schedulers.html。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● ram_allocation_ratio

配置选项 = 默认值	类型	Description
injected_network_template = \$pybasedir/nova/virt/interfaces.template	字符串值	<p>到 <code>/etc/network/interfaces</code> 模板的路径。</p> <p><code>/etc/network/interfaces-style</code> 文件的模板文件的路径，该文件将由 nova 填充，然后由 cloudinit 使用。这提供了一种在没有 DHCP 服务器的环境中配置网络连接的方法。</p> <p>模板将使用 Jinja2 模板引擎呈现，并接收名为 接口 的顶级密钥。此键将包含一个字典列表，每个接口对应一个。</p> <p>如需更多信息，请参阅 cloudinit 文档：</p> <p>https://cloudinit.readthedocs.io/en/latest/topics/datasources.html</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 到 Debian <code>/etc/network/interfaces</code> 文件的 Jinja2 格式的模板路径。即使使用非 Debian 派生的客户机也是如此。 <p>相关选项：</p> <ul style="list-style-type: none"> ● flat_inject：必须设置为 True，以确保 nova 将网络配置信息嵌入到通过配置驱动器提供的元数据中。
instance_build_timeout = 0	整数值	<p>实例构建可能需要的最长时间（以秒为单位）。</p> <p>如果此计时器过期，实例状态将更改为 ERROR。启用此选项可确保实例在较长的时间段内不会处于 BUILD 状态。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0：禁用选项（默认） ● 任何正整数（以秒为单位）：启用选项。

配置选项 = 默认值	类型	Description
instance_delete_interval = 300	整数值	<p>重试失败实例文件删除的时间间隔。</p> <p>这个选项取决于 <code>maximum_instance_delete_attempts</code>。此选项指定重试删除的频率，而 <code>maximum_instance_delete_attempts</code> 指定可以进行的最大重试尝试数。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0: 将以默认的定期间隔运行。 ● 任何值 < 0 : 禁用该选项。 ● 任何正整数（以秒为单位）。 <p>相关选项：</p> <ul style="list-style-type: none"> ● 来自 <code>instance_cleaning_opts</code> 组中的 maximum_instance_delete_attempts。
<code>`instance_format = [instance: %(uuid)s]`</code>	字符串值	通过日志消息传递的实例格式。
instance_name_template = instance-%08x	字符串值	<p>用于生成实例名称的模板字符串。</p> <p>此模板控制实例的数据库名称的创建。创建实例时，这不是 您输入的显示名称（通过 Horizon 或 CLI）。对于新部署，建议将默认值（使用数据库自动递增）更改为另一个值，它使用实例的属性，如 instance-%(uuid) s。如果您在更改它时已在部署中有实例，您的部署将中断。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 使用实例数据库 ID（如默认值）的字符串。 ● 包含命名数据库列的列表的字符串，如 %(id) d 或 %(uuid) s 或 %(hostname)。
instance_usage_audit = False	布尔值	此选项启用定期的 <code>compute.instance.exists</code> 通知。必须配置每个计算节点来生成系统使用量数据。这些通知由 OpenStack Telemetry 服务使用。

配置选项 = 默认值	类型	Description
instance_usage_audit_period = month	字符串值	<p>生成实例用量的时间周期。可以通过附加 @ 字符后跟一个数字定义偏移量来为给定周期定义可选偏移量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 周期，例如：小时、天、月份 或年 ● 带有偏移的时间，例如：month@15 将导致每月审计在每月 15 天开始。
<code>`instance_uuid_format = [instance:%(uuid)s]`</code>	字符串值	通过日志消息传递的实例 UUID 的格式。
instances_path = \$state_path/instances	字符串值	<p>指定实例存储在虚拟机监控程序磁盘上的位置。它可以指向本地附加存储或 NFS 上的目录。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● \$state_path/instances，其中 state_path 是配置选项，用于指定维护 nova 状态的顶级目录。（默认）或任何代表目录路径的字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [workarounds]/ensure_libvirt_rbd_instance_dir_cleanup
internal_service_availability_zone = internal	字符串值	<p>内部服务的可用区。</p> <p>此选项决定各种内部 nova 服务的可用区，如 nova-scheduler、nova-conductor 等。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表现有可用区名称的任何字符串。
key = None	字符串值	<p>SSL 密钥文件（如果与证书分开）。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● cert

配置选项 = 默认值	类型	Description
live_migration_retry_count = 30	整数值	live_migration 中最多 1 秒重试次数。它指定在 iptables 投诉时的重试次数。当用户持续将实时迁移请求发送到同一主机上时，会导致并发请求到 iptables。 可能的值： <ul style="list-style-type: none">任何代表重试计数的正整数。
log-config-append = None	字符串值	日志记录配置文件的名称。此文件附加到任何现有日志记录配置文件中。有关日志记录配置文件的详情，请查看 Python 日志记录模块文档。请注意，当使用日志记录配置文件时，所有日志记录配置都会在配置文件中设置，其他日志记录配置选项将被忽略（如 log-date-format）。
log-date-format = %Y-%m-%d %H:%M:%S	字符串值	在日志记录中定义 <code>%(asctime)s</code> 的格式字符串。默认： <code>%(default)s</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-dir = None	字符串值	（可选）用于相对 <code>log_file</code> 路径的基础目录。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log-file = None	字符串值	（可选）要将日志输出发送到的日志文件名称。如果没有设置默认值，日志记录将进入由 <code>use_stderr</code> 定义的 <code>stderr</code> 。如果设置了 <code>log_config_append</code> ，则忽略此选项。
log_options = True	布尔值	启动服务时，启用或禁用所有注册选项的日志值（在 <code>DEBUG</code> 级别）。
log_rotate_interval = 1	整数值	轮转日志文件前的时间长度。除非将 <code>log_rotation_type</code> 设置为 "interval"，否则此选项将被忽略。
log_rotate_interval_type = days	字符串值	轮转间隔类型。在调度下一个轮转时，将使用上次文件更改的时间（或服务启动的时间）。
log_rotation_type = none	字符串值	日志轮转类型。
logging_context_format_string = %(asctime)s.%(msecs)03d %(process) d %(levelname) s %(name) s [%(request_id) s %(user_identity) s] %(instance) s%(message) s	字符串值	使用上下文格式化用于日志消息的字符串。 <code>oslo_log.formatters.ContextFormatter</code> 使用

配置选项 = 默认值	类型	Description
logging_debug_format_s uffix = %(funcName)s % (pathname)s:%(lineno)d	字符串值	当消息的日志记录级别为 DEBUG 时，要附加到日志消息的其他数据。 oslo_log.formatters.ContextFormatter 使用
logging_default_format_s tring = %(asctime) s.% (msecs) 03d %(process) d %(levelname) s %(name) s [-] %(instance) s% (message) s	字符串值	上下文未定义时用于日志消息的字符串。 oslo_log.formatters.ContextFormatter 使用
logging_exception_prefix = %(asctime)s.% (msecs)03d %(process)d ERROR %(name)s % (instance)s	字符串值	用以下格式为每一行加前缀 异常输出。 oslo_log.formatters.ContextFormatter 使用
logging_user_identity_for mat = %(user) s %(tenant) s %(domain) s % (user_domain) s % (project_domain) s	字符串值	定义 logging_context_format_string 中使用的 % (user_identity) s 的格式字符串。 oslo_log.formatters.ContextFormatter 使用
long_rpc_timeout = 1800	整数值	<p>这个选项允许为可能需要很长时间的 RPC 调用设置备用超时值。如果设置，对其他服务的 RPC 调用将此值用于超时（以秒为单位），而不是全局 rpc_response_timeout 值。</p> <p>使用利用这个值的 RPC 调用操作：</p> <ul style="list-style-type: none"> ● 实时迁移 ● 调度 ● 启用/禁用计算服务 ● 镜像预缓存 ● 基于快照的/跨取消调整大小 ● 调整/冷迁移大小 ● 卷附加 <p>相关选项：</p> <ul style="list-style-type: none"> ● rpc_response_timeout

配置选项 = 默认值	类型	Description
max_concurrent_builds = 10	整数值	<p>限制 nova-compute 同时运行的最大实例构建数量。如果要求这样做，计算服务可以尝试构建无限数量的实例。实施此限制，以避免在计算节点上同时构建无限实例。此值可以为每个计算节点设置。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0: 被视为无限。 ● 任何代表并发构建的最大正整数。
max_concurrent_live_migrations = 1	整数值	<p>并发运行的最大实时迁移数量。这个限制是强制的，以避免出站实时迁移导致主机/网络造成问题。不建议您更改此设置，除非您确定环境中这样做是安全且稳定的。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0: 被视为无限。 ● 任何正整数，代表并发运行的最大实时迁移数。
max_concurrent_snapshots = 5	整数值	<p>同时运行的最大实例快照操作数。实施此限制以防止阻止主机/网络/存储造成故障的快照。此值可以为每个计算节点设置。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0: 被视为无限。 ● 任何正整数，代表最大并发快照。

配置选项 = 默认值	类型	Description
max_local_block_devices = 3	整数值	<p>在虚拟机监控程序节点上创建本地镜像的最大设备数。</p> <p>负数表示无限。将 max_local_block_devices 设置为 0 表示尝试创建本地磁盘的任何请求都将失败。这个选项旨在限制本地磁盘的数量（因此，root 本地磁盘是创建服务器时使用的 imageRef 的结果，以及任何其他临时磁盘和交换磁盘）。0 不意味着镜像将自动转换为卷并从卷引导实例，只是意味着尝试创建本地磁盘的所有请求都将失败。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0：不允许创建本地磁盘。 ● 负数：允许无限次本地磁盘。 ● 正数：只允许这些数量的本地磁盘。
max_logfile_count = 30	整数值	轮转日志文件的最大数量。
max_logfile_size_mb = 200	整数值	日志文件最大大小（以 MB 为单位）。如果 "log_rotation_type" 没有设置为 "size"，则忽略这个选项。
maximum_instance_delete_attempts = 5	整数值	<p>尝试获取实例文件的次数。</p> <p>这个选项指定可以进行的重试尝试的最大数量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数都定义了要进行多少尝试。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [DEFAULT] instance_delete_interval 可用于禁用这个选项。
metadata_listen = 0.0.0.0	字符串值	<p>元数据 API 将侦听的 IP 地址。</p> <p>元数据 API 服务侦听传入请求的此 IP 地址。</p>
metadata_listen_port = 8775	端口值	<p>元数据 API 将侦听的端口。</p> <p>元数据 API 服务侦听传入请求的此端口号。</p>

配置选项 = 默认值	类型	Description
metadata_workers = <based on operating system>	整数值	<p>元数据服务的 worker 数量。如果没有指定可用 CPU 的数量，则会使用。</p> <p>元数据服务可以配置为作为多进程(workers)运行。这解决了在 API 请求并发增加时降低吞吐量的问题。元数据服务将在指定的进程数中运行。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数 ● none（默认值）
migrate_max_retries = -1	整数值	<p>失败前重试实时迁移的次数。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● if == -1, 尝试直到主机不足（默认） ● 如果 == 0, 则只尝试一次, 则不会重试 ● 大于 0 的整数
mkisofs_cmd = genisoimage	字符串值	<p>用于 ISO 镜像创建的工具的名称或路径。</p> <p>使用 mkisofs_cmd 标志设置安装 genisoimage 程序的路径。如果 genisoimage 位于系统路径中, 则不需要更改默认值。</p> <p>要将配置驱动器与 Hyper-V 一起使用, 您必须将 mkisofs_cmd 值设置为 mkisofs.exe 安装的完整路径。另外, 您必须将 hyperv 配置部分中的 qemu_img_cmd 值设置为到 qemu-img 命令安装的完整路径。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● ISO 镜像创建程序的名称, 如果它与 nova-compute 服务位于同一个目录中 ● ISO 镜像创建程序的路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● 当启用配置驱动器时这个选项是有意义的。 ● 要在 Hyper-V 中使用配置驱动器, 您必须将 hyperv 配置部分中的 qemu_img_cmd 值设置为 qemu-img 命令安装的完整路径。

配置选项 = 默认值	类型	Description
my_block_storage_ip = \$my_ip	字符串值	<p>用于连接到块存储网络的 IP 地址。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 具有有效 IP 地址的字符串。默认为此主机的 IP 地址。 <p>相关选项：</p> <ul style="list-style-type: none"> ● my_ip - 如果没有设置 my_block_storage_ip, 则使用 my_ip 值。
my_ip = <based on operating system>	字符串值	<p>主机用于连接管理网络的 IP 地址。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 具有有效 IP 地址的字符串。默认为此主机的 IPv4 地址。 <p>相关选项：</p> <ul style="list-style-type: none"> ● my_block_storage_ip
network_allocate_retries = 0	整数值	<p>重试网络分配的次数。如果虚拟接口插件失败, 则需要尝试进行网络分配重试。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何代表重试计数的正整数。

配置选项 = 默认值	类型	Description
non_inheritable_image_properties = ['cache_in_nova', 'bittorrent']	列表值	<p>在进行快照时不应从实例继承的镜像属性。</p> <p>这个选项有机会选择新创建的快照不应继承哪个 image-properties。</p> <p>a. note::</p> <div style="border-left: 2px solid black; padding-left: 10px; margin-left: 20px;"> <p>The following image properties are <i>*never*</i> inherited regardless of whether they are listed in this configuration option or not:</p> <ul style="list-style-type: none"> ● cinder_encryption_key_id ● cinder_encryption_key_deletion_policy ● img_signature ● img_signature_hash_method ● img_signature_key_type ● img_signature_certificate_uuid </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● 以逗号分隔的列表，其项是镜像属性。通常，此处只能包含基础镜像所需的镜像属性，因为从基础镜像创建的快照不需要它们。 ● 默认列表：cache_in_nova, bittorrent
osapi_compute_listen = 0.0.0.0	字符串值	<p>OpenStack API 将侦听的 IP 地址。</p> <p>OpenStack API 服务侦听此 IP 地址以获取传入的请求。</p>
osapi_compute_listen_port = 8774	端口值	<p>OpenStack API 将侦听的端口。</p> <p>OpenStack API 服务侦听此端口号以获取传入的请求。</p>
<code>`osapi_compute_unique_server_name_scope = `</code>	字符串值	<p>设置检查唯一实例名称的范围。</p> <p>默认值不检查唯一名称。如果设置了名称检查的范围，则启动新实例或更新具有重复名称的现有实例将导致 <i>'InstanceExists'</i> 错误。唯一性区分大小写。设置此选项可以提高最终用户的可用性，因为他们不必通过 ID 来区分具有相同名称的实例。</p>

配置选项 = 默认值	类型	Description
osapi_compute_workers = None	整数值	<p>OpenStack API 服务的 worker 数量。默认值为可用的 CPU 数量。</p> <p>OpenStack API 服务可以配置为作为多进程(workers)运行。这解决了在 API 请求并发增加时降低吞吐量的问题。OpenStack API 服务将在指定的进程数中运行。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数 ● none（默认值）
password_length = 12	整数值	生成的实例管理员密码的长度。
periodic_enable = True	布尔值	<p>启用定期任务。</p> <p>如果设置为 true，则此选项允许服务在管理器中定期运行任务。</p> <p>如果运行多个调度程序或编排器，您可能只想在一个主机上运行定期任务 - 在这种情况下，请对所有主机禁用这个选项。</p>
periodic_fuzzy_delay = 60	整数值	<p>启动定期任务调度程序时随机延迟的秒数，以减少时间戳。</p> <p>当计算 worker 在集群中以 unison 重启时，它们最终都会同时运行定期任务，从而导致外部服务出现问题。要缓解此行为，periodic_fuzzy_delay 选项允许您在启动定期任务调度程序时引入随机初始延迟。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数（以秒为单位） ● 0：禁用随机延迟

配置选项 = 默认值	类型	Description
pointer_model = usbtouchpad	字符串值	<p>用于指定指针类型的通用属性。</p> <p>输入设备允许与图形帧缓冲进行交互。例如，要为绝对光标移动提供图形平板电脑。</p> <p>如果设置，hw_input_bus 或 hw_pointer_model 镜像元数据属性将优先于这个配置选项。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● usbtouchpad 必须在启用 VNC 的情况下配置，或者禁用 SPICE 代理。与 libvirt 一起使用时，实例模式应配置为 HVM。
preallocate_images = none	字符串值	<p>要使用的镜像预分配模式。</p> <p>镜像预分配允许在实例初始调配时预先分配实例镜像。这可确保如果没有足够的空间不可用，则会给出即时反馈。另外，它应该显著提高对新块写入的性能，甚至因为碎片降低，将 I/O 性能提高到预先写入的块。</p>
publish_errors = False	布尔值	<p>启用或禁用错误事件的发布。</p>
pybasedir = /usr/lib/python3.9/site- packages	字符串值	<p>安装 Nova python 模块的目录。</p> <p>此目录用于存储网络和远程控制台访问的模板文件。这也是需要保留 Nova 内部数据的其他配置选项的默认路径。不太可能需要修改这个选项的默认值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 到目录的完整路径。 <p>相关选项：</p> <ul style="list-style-type: none"> ● state_path

配置选项 = 默认值	类型	Description
ram_allocation_ratio = None	浮点值	<p>虚拟 RAM 与物理 RAM 分配比率。</p> <p>此选项用于通过配置 MEMORY_MB 清单的分配比率，来影响放置 API 选择的主机。</p> <p>a. note::</p> <div style="border-left: 2px solid black; padding-left: 10px; margin-left: 20px;"> <p>If this option is set to something *other than* `None` or `0.0`, the allocation ratio will be overwritten by the value of this option, otherwise, the allocation ratio will not change. Once set to a non-default value, it is not possible to "unset" the config to get back to the default behavior. If you want to reset back to the initial value, explicitly specify it to the value of `initial_ram_allocation_ratio`.</p> </div> <p>可能的值：</p> <ul style="list-style-type: none"> 任何有效的正整数或浮点值 <p>相关选项：</p> <ul style="list-style-type: none"> initial_ram_allocation_ratio
rate_limit_burst = 0	整数值	每个 rate_limit_interval 的最大日志信息数。
rate_limit_except_level = CRITICAL	字符串值	速率限制使用的日志级别名称：CRITICAL、ERROR、INFO、WARNING、DEBUG 或空字符串。大于或等于 rate_limit_except_level 的日志不会被过滤。空字符串表示过滤所有级别。
rate_limit_interval = 0	整数值	日志速率限制的间隔、秒数。
reboot_timeout = 0	整数值	<p>实例硬重启的时间间隔。</p> <p>当进行软重启时，客户机内核可能会完全挂起，这会导致软重启任务无法完成。如果将此选项设置为一段时间内（以秒为单位），如果实例一直处于重启状态超过 N 秒，则会自动硬重启实例。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 0：禁用选项（默认）。 任何正整数（以秒为单位）：启用选项。

配置选项 = 默认值	类型	Description
reclaim_instance_interval = 0	整数值	<p>重新声明已删除实例的间隔。</p> <p>大于 0 的值将启用 SOFT_DELETE 的实例。这个选项决定要删除的服务器是否被放入 SOFT_DELETED 状态。如果这个值大于 0，则不会立即删除删除的服务器，而是会被放入队列中，直到其太旧时间（删除的时间大于 reclaim_instance_interval 的值）。可以使用 restore 操作从删除队列中恢复服务器。如果已删除的服务器保持超过 reclaim_instance_interval 的值，它将由计算服务中的定期任务删除。</p> <p>请注意，这个选项是从 API 和计算节点读取的，且必须全局设置，否则服务器可以在 API 中放入软删除状态，永远不会实际回收（删除）计算节点上。</p> <p>a. 注意：使用此选项时，您还应配置 [cinder] 身份验证选项，例如 auth_type、auth_url、username 等。由于重新声明发生在定期任务中，因此没有用户令牌清理附加到任何 SOFT_DELETED 服务器的卷，因此 nova 必须配置有管理员角色访问权限来清理 cinder 中的这些资源。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何大于 0 的正整数（以秒为单位）将启用这个选项。 任何值 mvapich0 将禁用该选项。 <p>相关选项：</p> <ul style="list-style-type: none"> [Cinder] auth 选项用于在重新声明过程中清理附加到服务器的卷
record = None	字符串值	<p>用于存储此主机上运行的代理服务（如 VNC、spice、串行）的 websocket 帧的文件名。如果没有设置，则不会执行任何记录。</p>
report_interval = 10	整数值	<p>指示给定虚拟机监控程序上服务状态的频率（以秒为单位）。Nova 需要知道这一点以确定部署的整体健康状况。</p> <p>相关选项：</p> <ul style="list-style-type: none"> service_down_time report_interval 应该小于 service_down_time。如果 service_down_time 小于 report_interval，则服务通常会被视为 down，因为它们很少报告。

配置选项 = 默认值	类型	Description
rescue_timeout = 0	整数值	<p>在取消计算实例一直处于 RESCUE 前等待的时间。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0：禁用选项（默认） ● 任何正整数（以秒为单位）：启用选项。
reserved_host_cpus = 0	整数值	<p>为主机进程保留的主机 CPU 数量。</p> <p>主机资源使用情况从计算节点上运行的 nova-compute 持续报告回调度程序。这个值用于决定报告至放置的保留值。</p> <p>如果定义了 [compute] cpu_shared_set 或 [compute] cpu_dedicated_set 配置选项，则无法设置此选项。当定义这些选项时，这些值中没有包含的任何主机 CPU 都被视为为主机保留。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表要为主机保留的物理 CPU 数量的任何正整数。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [compute] cpu_shared_set ● [compute] cpu_dedicated_set
reserved_host_disk_mb = 0	整数值	<p>主机始终可用的磁盘资源量（以 MB 为单位）。磁盘用量报告的信息来自计算节点上运行的 nova-compute 的调度程序。要防止磁盘资源被视为可用，可以使用此选项为该主机保留磁盘空间。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数，代表主机保留的磁盘数量（以 MB 为单位）。
reserved_host_memory_mb = 512	整数值	<p>为主机保留的内存量（以 MB 为单位），使其始终可供主机进程使用。主机资源使用情况从计算节点上运行的 nova-compute 持续报告回调度程序。为防止主机内存被视为可用，可以使用此选项来为主机保留内存。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数，代表主机保留的内存量（以 MB 为单位）。

配置选项 = 默认值	类型	Description
reserved_huge_pages = None	dict 值	<p>每个 NUMA 主机单元保留的大内存页数量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 有效 key=value 列表，它反映了 NUMA 节点 ID，页面大小 <p>(默认单位为 KiB) 和要保留的页面数。例如：</p> <pre>reserved_huge_pages = node:0,size:2048,count:64 reserved_huge_pages = node:1,size:1GB,count:1</pre> <p>In this example we are reserving on NUMA node 0 64 pages of 2MiB and on NUMA node 1 1 page of 1GiB.</p>
resize_confirm_window = 0	整数值	<p>在 N 秒后自动确认调整大小。</p> <p>调整大小的功能将在调整大小前保存现有服务器。在调整大小完成后，要求用户确认调整大小。用户有机会确认或恢复所有更改。确认调整大小会删除原始服务器，并将服务器状态从 resized 改为 active。将此选项设置为时间段（以秒为单位）将自动确认服务器大小超过该时间。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 0：禁用选项（默认） 任何正整数（以秒为单位）：启用 选项。
resize_fs_using_block_device = False	布尔值	<p>启用通过块设备重新定义文件系统的大小。</p> <p>如果启用，尝试通过访问块设备的镜像来调整文件系统的大小。这由主机完成，如果镜像包含最新版本的 cloud-init，可能不需要这样做。可能的机制需要 nbd 驱动程序（对于 qcow 和 raw）或 loop（对于 raw）。</p>
resume_guests_state_on_host_boot = False	布尔值	<p>这个选项指定是否在主机重启前启动运行的虚拟机。它确保 Nova 计算节点上所有实例每次引导或重启时恢复其状态。</p>

配置选项 = 默认值	类型	Description
rootwrap_config = /etc/nova/rootwrap.conf	字符串值	<p>rootwrap 配置文件的路径。</p> <p>root 包装器的目标是允许特定于服务的非特权用户以安全的方式以 root 用户身份运行多个操作。此处使用的配置文件必须与 sudoers 条目中定义的配置文件匹配。</p>
rpc_conn_pool_size = 30	整数值	RPC 连接池的大小。
rpc_ping_enabled = False	布尔值	添加端点以回答 ping 调用。端点名为 oslo_rpc_server_ping
rpc_response_timeout = 60	整数值	从调用等待响应的秒数。
run_external_periodic_tasks = True	布尔值	某些定期任务可以在单独的进程中运行。我们是否应该在此处运行？
running_deleted_instance_action = reap	字符串值	<p>计算服务会定期检查数据库中已删除的实例，但仍然在计算节点上运行。以上选项支持在识别此类实例时采取行动。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● running_deleted_instance_poll_interval ● running_deleted_instance_timeout
running_deleted_instance_poll_interval = 1800	整数值	<p>运行之间等待的间隔（以秒为单位）进行清理操作。如果设置为 0，则禁用上述检查。如果 "running_deleted_instance_action" 设为 "log" 或 "reap"，则必须设置大于 0 的值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何正整数（以秒为单位）。 ● 0：禁用选项。 ● 1800: 默认值。 <p>相关选项：</p> <ul style="list-style-type: none"> ● running_deleted_instance_action

配置选项 = 默认值	类型	Description
running_deleted_instance_timeout = 0	整数值	<p>等待数据库中标记为已删除的实例的时间间隔（以秒为单位）有资格进行清理。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数（以秒为单位）。 <p>相关选项：</p> <ul style="list-style-type: none"> "running_deleted_instance_action"
scheduler_instance_sync_interval = 120	整数值	<p>发送调度程序当前实例 UUID 列表之间的间隔，以验证其实例视图是否与 nova 同步。</p> <p>如果 CONF 选项 <i>scheduler_tracks_instance_changes</i> 为 False，则不会进行同步调用。因此，更改这个选项将无效。</p> <p>如果同步情况不常见，则可能会增加这个间隔来降低发送的 RPC 信息数量。同样，如果同步问题出现问题，可以降低间隔来检查频率。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 0: 将以默认的定期间隔运行。 任何值 < 0 : 禁用该选项。 任何正整数（以秒为单位）。 <p>相关选项：</p> <ul style="list-style-type: none"> 如果 scheduler_tracks_instance_changes 设为 False，则此选项不会影响。
service_down_time = 60	整数值	<p>最后一次检查服务后的最大时间（以秒为单位）</p> <p>每个计算节点根据指定的报告间隔定期更新其数据库状态。如果计算节点没有更新超过 <i>service_down_time</i> 的状态，则计算节点将被视为 down。</p> <p>相关选项：</p> <ul style="list-style-type: none"> report_interval (service_down_time 不应小于 report_interval)

配置选项 = 默认值	类型	Description
servicegroup_driver = db	字符串值	<p>此选项指定用于 servicegroup 服务的驱动程序。</p> <p>nova 中的 ServiceGroup API 启用检查计算节点的状态。当运行 nova-compute 守护进程的 compute worker 启动时，它会调用 join API 来加入 compute 组。nova 调度程序等服务可以查询 ServiceGroup API 来检查节点是否处于活动状态。在内部，ServiceGroup 客户端驱动程序会自动更新计算 worker 状态。该服务有多个后端实现：Database ServiceGroup 驱动程序和 Memcache ServiceGroup 驱动程序。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● service_down_time（自上次检查以来为 up 服务提供最大时间）
shelved_offload_time = 0	整数值	<p>她的实例前的时间有资格从主机移除。</p> <p>默认情况下，此选项设置为 0，在 shelve 操作后立即从 hypervisor 中删除 shelved 实例。否则，实例将保留给 shelved_offload_time（以秒为单位）的值，以便在未shelve 操作的时间较快，然后在 shelved_offload_time 传递后，定期任务将从 hypervisor 中删除实例。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0: 实例将在 shelve 后立即卸载。 ● 任何值 < 0: 一个实例永远不会卸载。 ● 任何正整数（以秒为单位）：实例将在卸载前指定的秒数。
shelved_poll_interval = 3600	整数值	<p>轮询 sheed 实例以卸载的间隔。</p> <p>定期任务运行每个 shelved_poll_interval 数的秒数，并检查是否有 shelved 实例。如果找到一个 shelved 实例，基于 shelved_offload_time 配置值，它会卸载 shelved 实例。检查 shelved_offload_time 配置选项描述以了解详细信息。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● any value criu 0：禁用选项。 ● 任何正整数（以秒为单位）。 <p>相关选项：</p> <ul style="list-style-type: none"> ● shelved_offload_time

配置选项 = 默认值	类型	Description
shutdown_timeout = 60	整数值	<p>实例执行清理关闭的总时间（以秒为单位）。</p> <p>它决定虚拟机执行干净的整个期间（以秒为单位）。在执行 stop、rescue 和 shelve 重建操作时，配置此选项可让虚拟机在实例关闭前执行受控关闭。默认超时时间为 60 秒。值 0（零）表示客户机将立即关闭，且对 guest OS 清理没有机会。</p> <p>超时值可以基于每个镜像覆盖，方法为 <code>os_shutdown_timeout</code>，它是一个镜像元数据设置，允许不同类型的操作系统指定需要完全关闭的时间。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数或 0（默认值为 60）。
source_is_ipv6 = False	布尔值	<p>如果源主机使用 IPv6 寻址，则设置为 True。</p>
ssl_only = False	布尔值	<p>不允许非加密连接。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● cert ● key
state_path = \$pybasedir	字符串值	<p>维护 Nova 状态的顶级目录。</p> <p>此目录用于存储 Nova 的内部状态。它被从此派生的不同其他配置选项使用。在某些情况下（如迁移）使用在多个计算主机（例如通过 NFS）之间共享的存储位置有意义。除非选项 <code>instances_path</code> 被覆盖，否则此目录可能会变得非常大。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 到目录的完整路径。默认为 <code>pybasedir</code> 中提供的值。

配置选项 = 默认值	类型	Description
sync_power_state_interval = 600	整数值	<p>在数据库和虚拟机监控程序之间同步电源状态的间隔。</p> <p>Nova 检查实际虚拟机电源状态和 Nova 在其数据库中的电源状态的时间间隔。如果用户关闭其虚拟机，Nova 会更新 API 以报告虚拟机已关闭。如果意外打开虚拟机，Nova 将关闭虚拟机，使系统处于预期状态。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0: 将以默认的定期间隔运行。 ● 任何值 < 0 : 禁用该选项。 ● 任何正整数（以秒为单位）。 <p>相关选项：</p> <ul style="list-style-type: none"> ● 如果 workarounds 组中的 handle_virt_lifecycle_events 为 false，且此选项为负数，那么管理程序和 Nova 数据库之间的不同步的实例必须手动同步。
sync_power_state_pool_size = 1000	整数值	<p>可用于同步电源状态的绿色线程数。</p> <p>此选项可用于减少对 hypervisor 或系统发出的并发请求数量，并出于性能原因（例如，使用 Ironic）的实际实例电源状态。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表绿色线程数的任何正整数。
syslog-log-facility = LOG_USER	字符串值	syslog 工具接收日志行。如果设置了 log_config_append，则忽略此选项。
tempdir = None	字符串值	明确指定临时工作目录。
timeout_nbd = 10	整数值	等待 NBD 设备启动的时间（以秒为单位）。

配置选项 = 默认值	类型	Description
transport_url = rabbit://	字符串值	<p>用于连接到消息传递后端的网络地址和可选用户凭证，采用 URL 格式。预期格式为：</p> <pre>driver://[user:pass@]host:port[, [userN:passN@]hostN:portN]/virtual_host?query</pre> <p>示例：rabbit://rabbitmq:password@127.0.0.1:5672//</p> <p>有关 URL 中字段的详情，请查看 <code>oslo_messaging.TransportURL</code> 文档 https://docs.openstack.org/oslo.messaging/latest/reference/transport.html</p>
update_resources_interval = 0	整数值	<p>更新计算资源的间隔。</p> <p>这个选项指定 <code>update_available_resource periodic</code> 任务应运行的频率。小于 0 的值表示完全禁用任务。保留它的默认值 0 将导致以默认的定期间隔运行。把它设置为任何正值将导致其在大约该秒数中运行。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0: 将以默认的定期间隔运行。 ● 任何值 < 0 : 禁用该选项。 ● 任何正整数（以秒为单位）。
use-journal = False	布尔值	<p>为日志记录启用 <code>journald</code>。如果在 <code>systemd</code> 环境中运行，您可能需要启用日志支持。这样做将使用日志原生协议，该协议除日志消息之外还包含结构化元数据。如果设置了 <code>log_config_append</code>，此选项将被忽略。</p>
use-json = False	布尔值	<p>使用 JSON 格式进行日志记录。如果设置了 <code>log_config_append</code>，则忽略此选项。</p>
use-syslog = False	布尔值	<p>使用 <code>syslog</code> 进行日志记录。现有的 <code>syslog</code> 格式是 DEPRECATED，稍后将更改为遵循 RFC5424。如果设置了 <code>log_config_append</code>，则忽略此选项。</p>
use_cow_images = True	布尔值	<p>启用使用写时复制(cow)镜像。</p> <p>QEMU/KVM 允许使用 <code>qcow2</code> 作为后备文件。禁用此功能，将不会使用后备文件。</p>
use_eventlog = False	布尔值	<p>日志输出到 Windows 事件日志。</p>

配置选项 = 默认值	类型	Description
use_rootwrap_daemon = False	布尔值	启动并启用可运行需要以 root 特权运行的命令的守护进程。此选项通常在运行 nova 计算进程的节点上启用。
use_stderr = False	布尔值	将输出输出到标准错误。如果设置了 log_config_append，则忽略此选项。
vcpu_pin_set = None	字符串值	<p>可用于 VCPU 资源的主机 CPU 掩码。</p> <p>这个选项的行为取决于 [compute] cpu_dedicated_set 选项的定义，并影响 [compute] cpu_shared_set 选项的行为。</p> <ul style="list-style-type: none"> ● 如果定义了 [compute] cpu_dedicated_set，定义这个选项将导致错误。 ● 如果没有定义 [compute] cpu_dedicated_set，这个选项将用于决定 VCPU 资源的清单，并限制固定和未固定实例可以调度到的主机 CPU，覆盖 [compute] cpu_shared_set 选项。 <p>可能的值：</p> <ul style="list-style-type: none"> ● 以逗号分隔的物理 CPU 号列表，虚拟 CPU 可以从中分配。每个元素应该是单个 CPU 号、一个 CPU 数字范围，或者后面有一个要排除的 CPU 号。 <p>前一个范围。例如：</p> <pre>vcpu_pin_set = "4-12,^8,15"</pre> <p>相关选项：</p> <ul style="list-style-type: none"> ● [compute] cpu_dedicated_set ● [compute] cpu_shared_set <p>弃用自：20.0.0</p> <p>原因： 此选项已被 "[compute] cpu_dedicated_set" 和 "[compute] cpu_shared_set" 选项替代，它允许同一主机上固定和未固定实例（用于 libvirt 驱动程序）等内容。</p>

配置选项 = 默认值	类型	Description
vif_plugging_is_fatal = True	布尔值	<p>确定实例是否应该在 VIF 插入超时上引导或失败。</p> <p>Nova 在实例调度后向 Neutron 发送端口更新，为 Neutron 提供必要信息来完成端口的设置。完成后，Neutron 会通知 Nova 已完成设置端口，此时 Nova 恢复实例的引导，因为现在应当存在网络连接。如果在给定间隔后没有收到回复，则会出现超时。</p> <p>此选项决定发生 VIF 插入超时事件时 Nova 的作用。启用后，实例将出错。禁用后，实例将继续在端口就绪时引导。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：在 VIF 插入超时后实例应该会失败 ● false：实例应在 VIF 插入超时后继续引导
vif_plugging_timeout = 300	整数值	<p>Neutron VIF 插入事件消息的超时。</p> <p>等待 Neutron vif 插入事件进入或失败的秒数（请参阅 <i>vif_plugging_is_fatal</i>）。</p> <p>如果您大规模遇到超时失败，请考虑通过 [agent]/root_helper_daemon neutron 配置选项在 neutron 代理中运行 rootwrap in "daemon mode"。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● vif_plugging_is_fatal - 如果 vif_plugging_timeout 被设置为零，vif_plugging_is_fatal 为 False，则事件不应该被完全到达。
virt_mkfs = []	多值	<p>临时设备的 mkfs 命令的名称。</p> <p>格式为 <os_type>=<mkfs 命令></p>
volume_usage_poll_interval = 0	整数值	<p>收集卷使用情况的时间间隔。</p> <p>这个选项为每个 volume_usage_poll_interval 秒数更新卷使用缓存。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何大于 0 的正整数（以秒为单位）将启用这个选项。 ● 任何值 mvapich0 将禁用该选项。

配置选项 = 默认值	类型	Description
watch-log-file = False	布尔值	使用旨在监视文件系统的日志记录处理程序。移动或删除日志文件时，此处理程序将立即打开具有指定路径的新日志文件。只有在指定了 <code>log_file</code> 选项并使用 Linux 平台时，才有意义。如果设置了 <code>log_config_append</code> ，则忽略此选项。
web = /usr/share/spice-html5	字符串值	包含将由 Web 服务器提供的内容的目录路径。

9.1.2. api

下表概述了 `/etc/nova/nova.conf` 文件中的 `[api]` 组下可用的选项。

表 9.1. api

配置选项 = 默认值	类型	Description
auth_strategy = keystone	字符串值	<p>确定用于身份验证的策略。</p> <p>弃用自： 21.0.0</p> <p>原因： 唯一非默认选择 "noauth2"，仅用于内部开发和测试目的，不应在部署中使用。这个选项及其中间件 <code>NoAuthMiddleware[V2_18]</code> 将在以后的版本中删除。</p>
compute_link_prefix = None	字符串值	<p>此字符串前面带有到 OpenStack Compute API 链接返回的普通 URL。如果为空（默认），则 URL 会以保持不变。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何字符串，包括空字符串（默认）。

配置选项 = 默认值	类型	Description
config_drive_skip_versions = 1.0 2007-01-19 2007-03-01 2007-08-29 2007-10-10 2007-12-15 2008-02-01 2008-09-01	字符串值	<p>当为配置驱动器收集现有元数据时，会为没有出现在这个选项中的所有版本返回 EC2-style 元数据。从 crun 发行版本开始，可用版本有：</p> <ul style="list-style-type: none"> ● 1.0 ● 2007-01-19 ● 2007-03-01 ● 2007-08-29 ● 2007-10-10 ● 2007-12-15 ● 2008-02-01 ● 2008-09-01 ● 2009-04-04 <p>选项采用单个字符串的格式，每个版本用空格分开。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表零个或多个版本的字符串，用空格分开。
dhcp_domain = novalocal	字符串值	<p>用于为实例配置 FQDN 的域名。</p> <p>为实例主机名配置完全限定域名。如果未设置，则只配置没有域的主机名。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何作为有效域名的字符串。
enable_instance_password = True	布尔值	<p>启用通过相关服务器 API 调用返回实例密码，如创建、重建、撤离或救援。如果虚拟机监控程序不支持密码注入，则返回的密码将不正确，因此如果您的虚拟机监控程序不支持密码注入，则将其设置为 False。</p>
glance_link_prefix = None	字符串值	<p>此字符串前面带有到 Glance 资源链接返回的普通 URL。如果为空（默认），则 URL 会以保持不变。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何字符串，包括空字符串（默认）。

配置选项 = 默认值	类型	Description
instance_list_cells_batch_fixed_size = 100	整数	<p>如果 instance_list_cells_batch_strategy 设置为 固定，则控制从每个单元数据库请求的实例的批处理大小。此完整值将定义在每次请求批处理实例时为每个单元发出的限制，无论系统中的单元数或其他因素如何。根据 instance_list_cells_batch_strategy 文档中的调用的一般逻辑，每个批处理的最小值是 100 个记录。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● instance_list_cells_batch_strategy ● max_limit
instance_list_cells_batch_strategy = distributed	字符串	<p>这控制 API 在大型实例列表操作过程中以较小的批处理查询单元数据库的方法。如果执行批处理，则大型实例列表操作将首先从每个单元格数据库中请求部分整体 API 限制，并且会重新请求与每个单元消耗相同的批处理大小（根据需要从每个单元中消耗（返回）。更大的批处理意味着 API 和数据库之间的 chattiness 较少，但可能会处理不会向用户返回的数据库的结果。任何策略将生成至少 100 个记录的批处理大小，以避免用户在请求中造成多个小数据库查询。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● instance_list_cells_batch_fixed_size ● max_limit
instance_list_per_project_cells = False	布尔值	<p>启用后，这将导致 API 仅查询租户映射实例的单元数据库。这要求在每个列表前在 API 数据库中进行额外的(fast)查询，但也（可能）限制必须查询的单元数据库数量以提供结果。如果您有少量单元，或者租户可能在所有单元中都有实例，则这应该是 False。如果您有很多单元，特别是如果您将租户限制为这些单元的一小部分，则这应该是 True。</p>

配置选项 = 默认值	类型	Description
list_records_by_skipping_down_cells = True	布尔值	<p>当设置为 False 时，如果基础架构故障（如不响应的单元格）时，这会导致 API 返回一个 500 错误。如果您希望 API 跳过 down 单元，并从 up cells 设置这个选项设置为 True 时返回结果。</p> <p>请注意，在 API microversion 2.69 中，部署某些记录不可用时可能会出现临时条件，结果可能部分用于包含这些记录的某些请求。在这种情况下，这个选项将被忽略。如需更多信息，请参阅 Compute API 指南中的“处理关闭 Cells”部分 (https://docs.openstack.org/api-guide/compute/down_cells.html)。</p>
local_metadata_per_cell = False	布尔值	<p>表示 nova-metadata API 服务已按原样部署，以便在多 cell 部署中拥有更好的性能和数据隔离。根据 neutron 的设置方式，用户应考虑使用此配置。如果您有跨越单元格的网络，可能需要全局运行 nova-metadata API 服务。如果您的网络分段在单元格内，您可以按单元运行 nova-metadata API 服务。当每个单元运行 nova-metadata API 服务时，您还应配置每个 Neutron metadata-agent 以指向对应的 nova-metadata API 服务。</p>
max_limit = 1000	整数值	<p>作为查询可能会返回数以千计的项目，您可以通过设置这个选项来限制单个响应中的最大项目数。</p>
metadata_cache_expiration = 15	整数值	<p>这个选项是缓存元数据的时间（以秒为单位）。当设置为 0 时，元数据缓存会被完全禁用，通常不建议出于性能考虑。增加此设置应该可在负载过重时提高元数据 API 的响应时间。数值越高，可能会提高内存用量，导致主机元数据更改生效的时间更长。</p>
neutron_default_tenant_id = default	字符串值	<p>用于从 Neutron API 获取默认网络的租户 ID（也称为项目 ID）。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● use_neutron_default_nets
use_forwarded_for = False	布尔值	<p>为 True 时，X-Forwarded-For 标头被视为规范远程地址。当 False（默认值）时，会使用 remote_address 标头。</p> <p>只有在您有一个 HTML sanitizing 代理时才应启用它。</p>

配置选项 = 默认值	类型	Description
use_neutron_default_networks = False	布尔值	<p>为 True 时，TenantNetworkController 将查询 Neutron API 来获取要使用的默认网络。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● neutron_default_tenant_id
vendordata_dynamic_connect_timeout = 5	整数值	<p>外部 REST 服务连接的最大等待时间。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何值大于 3 的整数 (TCP 数据包重新传输超时)。请注意，这个等待时间可能会阻断实例启动，因此这个值应该保持小。 <p>相关选项：</p> <ul style="list-style-type: none"> ● vendordata_providers ● vendordata_dynamic_targets ● vendordata_dynamic_ssl_certfile ● vendordata_dynamic_read_timeout ● vendordata_dynamic_failure_fatal
vendordata_dynamic_failure_fatal = False	布尔值	<p>获取动态 vendordata 失败是否对实例启动是致命的？</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● vendordata_providers ● vendordata_dynamic_targets ● vendordata_dynamic_ssl_certfile ● vendordata_dynamic_connect_timeout ● vendordata_dynamic_read_timeout

配置选项 = 默认值	类型	Description
vendordata_dynamic_read_timeout = 5	整数值	<p>外部 REST 服务在连接后返回数据的最大等待时间。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 任何整数。请注意，在这个等待时间，实例启动会被阻断，因此这个值应该保持小。 <p>相关选项：</p> <ul style="list-style-type: none"> ● vendordata_providers ● vendordata_dynamic_targets ● vendordata_dynamic_ssl_certfile ● vendordata_dynamic_connect_timeout ● vendordata_dynamic_failure_fatal
<code>vendordata_dynamic_ssl_certfile = `</code>	字符串值	<p>可选的证书文件或 CA 捆绑包的路径，用于验证动态厂商数据 REST 服务 ssl 证书。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个空字符串，或到有效证书文件的路径 <p>相关选项：</p> <ul style="list-style-type: none"> ● vendordata_providers ● vendordata_dynamic_targets ● vendordata_dynamic_connect_timeout ● vendordata_dynamic_read_timeout ● vendordata_dynamic_failure_fatal
vendordata_dynamic_targets = []	列表值	<p>动态 vendordata 供应商的目标列表。这些目标的格式是 <code><name>@<url></code>。</p> <p>动态 vendordata 提供程序通过联系外部 REST 服务来收集元数据，并查询它们以获取有关实例的信息。此行为记录在 nova 开发人员引用的 vendordata.rst 文件中。</p>

配置选项 = 默认值	类型	Description
vendordata_jsonfile_path = None	字符串值	<p>云供应商可以将自定义数据存储于厂商数据文件中，然后通过元数据服务提供给实例，以及 config-drive 的渲染。此默认类 JsonFileVendorData 从 JSON 文件加载此信息，该文件路径由此选项配置。如果此选项没有设置路径，则类会返回一个空的字典。</p> <p>请注意，当使用它来向配置驱动器提供静态厂商数据时，必须配置 nova-compute 服务，且该文件必须从 nova-compute 主机访问。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 代表数据文件的路径或空字符串（默认）的任何字符串。
vendordata_providers = ['StaticJSON']	列表值	<p>vendordata 供应商列表。</p> <p>vendordata 供应商是通过特定于其部署的 configdrive 和元数据提供元数据。</p> <p>有关实施 vendordata 动态端点的要求的更多信息，请参阅 nova 开发人员参考中的 vendordata.rst 文件。</p> <p>相关选项：</p> <ul style="list-style-type: none"> vendordata_dynamic_targets vendordata_dynamic_ssl_certfile vendordata_dynamic_connect_timeout vendordata_dynamic_read_timeout vendordata_dynamic_failure_fatal

9.1.3. api_database

下表概述了 `/etc/nova/nova.conf` 文件中的 `[api_database]` 组下可用的选项。

表 9.2. api_database

配置选项 = 默认值	类型	Description
connection = None	字符串值	用于连接到数据库的 SQLAlchemy 连接字符串。不要为 nova-compute 服务设置它。
connection_debug = 0	整数值	SQL 调试信息的详细程度：0=None, 100=Everything。

配置选项 = 默认值	类型	Description
<code>`connection_parameters = `</code>	字符串值	用于连接时附加到连接 URL 的可选 URL 参数；指定 <code>param1=value1&param2=value2&...</code>
connection_recycle_time = 3600	整数值	连接池中存在的连接的时间超过这个秒数，下一次从池中签出它们的时间将替换为新的连接。
connection_trace = False	布尔值	将 Python 堆栈跟踪作为注释字符串添加到 SQL。
max_overflow = None	整数值	如果设置，请将这个值用于 SQLAlchemy 的 <code>max_overflow</code> 。
max_pool_size = None	整数值	在池中保留的最大 SQL 连接数。将值设为 0 表示没有限制。
max_retries = 10	整数值	启动期间数据库连接重试的最大数量。设置为 -1 以指定无限重试计数。
mysql_sql_mode = TRADITIONAL	字符串值	用于 MySQL 会话的 SQL 模式。这个选项（包括默认选项）会覆盖任何 <code>server-set</code> SQL 模式。要使用服务器配置设置的任何 SQL 模式，请将其设置为 <code>no</code> 值。示例： <code>mysql_sql_mode=</code>
pool_timeout = None	整数值	如果设置，请将这个值用于 SQLAlchemy 的 <code>pool_timeout</code> 。
retry_interval = 10	整数值	重新尝试打开 SQL 连接之间的间隔。
slave_connection = None	字符串值	用于连接到从数据库的 SQLAlchemy 连接字符串。
sqlite_synchronous = True	布尔值	如果为 <code>True</code> ，则 SQLite 使用同步模式。

9.1.4. barbican

下表概述了 `/etc/nova/nova.conf` 文件中的 `[barbican]` 组下可用的选项。

表 9.3. barbican

配置选项 = 默认值	类型	Description
auth_endpoint = http://localhost/identity/v3	字符串值	使用此端点连接到 Keystone
barbican_api_version = None	字符串值	Barbican API 的版本，例如： <code>"v1"</code>

配置选项 = 默认值	类型	Description
barbican_endpoint = None	字符串值	使用此端点连接到 Barbican，例如： "http://localhost:9311/"
barbican_endpoint_type = public	字符串值	指定端点的类型。允许的值有：public、private 和 admin
number_of_retries = 60	整数值	在创建密钥时重试轮询的次数
retry_delay = 1	整数值	重试轮询创建密钥完成前等待的秒数
verify_ssl = True	布尔值	指定是否不安全的 TLS (https) 请求。如果为 False，则不会验证服务器证书，如果为 True，则可以同时设置 verify_ssl_path 配置。
verify_ssl_path = None	字符串值	要检查的捆绑包或 CA 证书的路径，或 None 请求尝试查找和使用 verify_ssl 为 True 的证书。如果 verify_ssl 是 False，这将被忽略。

9.1.5. 缓存

下表概述了 `/etc/nova/nova.conf` 文件中的 `[cache]` 组下可用的选项。

表 9.4. 缓存

配置选项 = 默认值	类型	Description
backend = dogpile.cache.null	字符串值	缓存后端模块。对于具有数百个线程服务器的基于 eventlet 的环境，建议使用带有池 (oslo_cache.memcache_pool) 的 Memcache。对于小于 100 个线程服务器的环境，建议使用 Memcached (dogpile.cache.memcached) 或 Redis (dogpile.cache.redis)。具有单一服务器实例的测试环境可以使用 dogpile.cache.memory 后端。
backend_argument = []	多值	为 backend 模块提供的参数。指定每参数传递到 dogpile.cache 后端的一次这个选项。示例格式： " <argname>:<value>"。
config_prefix = cache.oslo	字符串值	为缓存区域构建配置字典的前缀。除非存在另一个具有相同配置名称的 dogpile.cache 区域，否则这不应更改。
dead_timeout = 60	浮点值	尝试在 HashClient 内部机制中重新添加节点前的时间 (以秒为单位)。

配置选项 = 默认值	类型	Description
debug_cache_backend = False	布尔值	从缓存后端进行额外的调试（缓存键、get/set/delete/etc 调用）。只有在您需要使用键/值查看特定的 cache-backend get/set/delete 调用时才有用。通常，这应该设为 false。
enable_retry_client = False	布尔值	启用重试客户端机制来处理失败。这些机制可用于嵌套所有的 pymemcache 客户端。打包程序允许您定义有多少尝试进行，以及 attempts 之间等待的时间。
enable_socket_keepalive = False	布尔值	dogpile 的 pymemcache 后端的套接字 keepalive 的全局切换
enabled = False	布尔值	缓存的全局切换。
expiration_time = 600	整数值	dogpile.cache 区域中的任何缓存项的默认 TTL（以秒为单位）。这适用于任何未为其定义显式缓存过期时间的缓存方法。
hashclient_retry_attempts = 2	整数值	在客户端标记为死前应尝试的次数，并在 HashClient 的内部机制中从池中删除。
hashclient_retry_delay = 1	浮点值	在 HashClient 内部机制中重试尝试之间应经过的时间（以秒为单位）。
memcache_dead_retry = 300	整数值	在再次尝试前，memcached 服务器被视为 dead 的秒数。（dogpile.cache.memcache 和 oslo_cache.memcache_pool 后端）。
memcache_pool_connection_get_timeout = 10	整数值	操作将等待的秒数，以获取 memcache 客户端连接。
memcache_pool_flush_on_reconnect = False	布尔值	如果 memcache 在重新连接时清除，则全局切换。（仅oslo_cache.memcache_pool 后端）。
memcache_pool_maxsize = 10	整数值	每个 memcached 服务器的最大打开连接数。（oslo_cache.memcache_pool backend）。
memcache_pool_unused_timeout = 60	整数值	到 memcached 的连接在池关闭前没有被使用的秒数。（仅oslo_cache.memcache_pool 后端）。

配置选项 = 默认值	类型	Description
memcache_servers = ['localhost:11211']	列表值	memcache 服务器格式为 "host:port"。 (dogpile.cache.memcached 和 oslo_cache.memcache_pool 后端)。如果给定主机引用 IPv6 或给定域，则您应该为给定地址系列(inet6) (inet6) (inet6) (inet6[::1]:11211,inete6:[fd12:3456:789a:1::1]:11211,inete6:[controller-0.internalapi]:11211)作为前缀。如果未提供地址系列，则使用的默认地址系列将是 inet (对应于 IPv4)
memcache_socket_timeout = 1.0	浮点值	每次对服务器调用的超时时间 (以秒为单位)。 (dogpile.cache.memcache 和 oslo_cache.memcache_pool 后端)。
proxies = []	列表值	用于导入的代理类会影响 dogpile.cache 后端功能的方式。有关 change-backend-behavior 的 dogpile.cache 文档。
retry_attempts = 2	整数值	在失败前尝试操作的次数。
retry_delay = 0	浮点值	每次尝试之间休眠的秒数。
socket_keepalive_count = 1	整数值	在丢弃连接前，保持 TCP 的最大 keepalive 探测 TCP 数量。应该是大于零的正整数。
socket_keepalive_idle = 1	整数值	在 TCP 开始发送 keepalive 探测前，连接需要保持闲置的时间 (以秒为单位)。应该是大于零的正整数。
socket_keepalive_interval = 1	整数值	各个 keepalive 探测之间的时间 (以秒为单位)。应该是大于零的正整数。
tls_allowed_ciphers = None	字符串值	为使用 TLS 上下文创建的套接字设置可用密码。它应该是 OpenSSL 密码列表格式的字符串。如果没有指定，则所有启用了 OpenSSL 的密码都可用。
tls_cafile = None	字符串值	以 PEM 格式串联的 CA 证书文件的路径，以建立缓存服务器的真实性。如果 <code>tls_enabled</code> 是 <code>False</code> ，这个选项将被忽略。
tls_certfile = None	字符串值	到 PEM 格式的单个文件的路径，其中包含客户端证书以及建立证书真实性所需的任意数量的 CA 证书。只有在需要客户端身份验证时才需要此文件。如果 <code>tls_enabled</code> 是 <code>False</code> ，这个选项将被忽略。
tls_enabled = False	布尔值	与缓存服务器组合时，TLS 使用的全局切换。

配置选项 = 默认值	类型	Description
tls_keyfile = None	字符串值	到包含客户端私钥的单个文件的路径。其他私钥将从 <code>tls_certfile</code> 中指定的文件获取。如果 <code>tls_enabled</code> 是 <code>False</code> ，这个选项将被忽略。

9.1.6. cinder

下表概述了 `/etc/nova/nova.conf` 文件中的 `[cinder]` 组下可用的选项。

表 9.5. cinder

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
catalog_info = volumev3::publicURL	字符串值	<p>在服务目录中查找 <code>cinder</code> 时匹配的信息。</p> <p>& <code>lt;service_name</code> > 是可选的，默认省略，因为大多数部署中不需要它。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 格式是分开的值：<code><service_type>:<service_name>:<endpoint_type></code> <p>注意：自 Nova Nova 17.0.0 Queens 版本，将不再支持 Cinder v2 API。</p> <p>相关选项：</p> <ul style="list-style-type: none"> <code>endpoint_template</code> - 设置此选项将覆盖 <code>catalog_info</code>
certFile = None	字符串值	PEM 编码的客户端证书证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。

配置选项 = 默认值	类型	Description
cross_az_attach = True	布尔值	<p>允许在不同可用区中的实例和卷之间附加。</p> <p>如果 False，与 Nova 中的实例可用区的 Cinder 中必须位于同一可用区中的卷。</p> <p>这也意味着从源不是“卷”的卷引导实例时应小心，因为 Nova 将尝试使用与分配给实例相同的可用区创建卷。</p> <p>如果 AZ 不在 Cinder 中（或者 cinder.conf 中的 allow_availability_zone_fallback=False），则卷创建请求将失败，实例将失败。</p> <p>默认情况下，卷附加没有可用区限制。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● [DEFAULT]/default_schedule_zone
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
endpoint_template = None	字符串值	<p>如果设置了这个选项，它将使用 cinder 端点的此模板覆盖服务目录查找</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● cinder 端点 API 的 URL，例如 http://localhost:8776/v3/%(project_id)s <p>注意：自 Nova Nova 17.0.0 Queens 版本，将不再支持 Cinder v2 API。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● catalog_info - 如果没有设置 endpoint_template，则将使用 catalog_info。

配置选项 = 默认值	类型	Description
http_retries = 3	整数值	cinderclient 在任何失败的 http 调用中重试的次数。0 表示只尝试一次连接。把它设置为任何正整数表示在失败连接中重试多次，例如将其设置为 3 表示连接总数为 4。 可能的值： <ul style="list-style-type: none">任何整数值。0 表示只尝试一次连接
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
os_region_name = None	字符串值	此节点的区域名称。这在选择服务目录中的 URL 时使用。 可能的值： <ul style="list-style-type: none">代表区域名称的任何字符串
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
split-loggers = False	布尔值	将请求记录到多个日志记录器。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID

配置选项 = 默认值	类型	Description
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名

9.1.7. Compute

下表概述了 `/etc/nova/nova.conf` 文件中的 `[compute]` 组下可用的选项。

表 9.6. Compute

配置选项 = 默认值	类型	Description
consecutive_build_service_disable_threshold = 10	整数值	<p>启用向调度程序报告构建失败。</p> <p>任何非零值都将启用向调度程序发送构建失败统计信息，供 BuildFailureWeigher 使用。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数都允许报告构建失败。 零以禁用报告构建失败。 <p>相关选项：</p> <ul style="list-style-type: none"> <code>[filter_scheduler]/build_failure_weight_multiplier</code>

配置选项 = 默认值	类型	Description
cpu_dedicated_set = None	字符串值	<p>可用于 PCPU 资源的主机 CPU 掩码。</p> <p>此选项的行为会影响已弃用的 vcpu_pin_set 选项的行为。</p> <ul style="list-style-type: none"> ● 如果定义了这个选项，定义 vcpu_pin_set 将导致错误。 ● 如果没有定义这个选项，则使用 vcpu_pin_set 来确定 VCPU 资源的清单，并限制固定和未固定实例可以调度到的主机 CPU。 <p>当删除 vcpu_pin_set 时，将在以后的版本中简化此行为。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 以逗号分隔的物理 CPU 号列表，实例 VCPU 可以从中分配。每个元素应该是单个 CPU 号、一个 CPU 数字范围，或者后面有一个要排除的 CPU 号。 <p>前一个范围。例如：</p> <pre>cpu_dedicated_set = "4-12,^8,15"</pre> <p>相关选项：</p> <ul style="list-style-type: none"> ● [compute] cpu_shared_set：这是定义应从中分配 VCPU 资源的位置的对应选项。 ● vcpu_pin_set：此选项部分替换的传统选项。

配置选项 = 默认值	类型	Description
cpu_shared_set = None	字符串值	<p>用于 VCPU 资源和卸载仿真程序线程的主机 CPU 掩码。</p> <p>这个选项的行为取决于已弃用的 vcpu_pin_set 选项的定义。</p> <ul style="list-style-type: none"> ● 如果没有定义 vcpu_pin_set, 则使用 [compute] cpu_shared_set 提供 VCPU 清单, 并确定未固定实例可以调度到的主机 CPU。它还将用于确定应该为使用共享仿真器线程策略 (hw:emulator_threads_policy= share) 配置的实例卸载到的主机 CPUS。 ● 如果定义了 vcpu_pin_set, 则 [compute] cpu_shared_set 将只用于决定主机 CPU, 仿真程序线程应该被卸载到使用 share 仿真程序线程策略 (hw:emulator_threads_policy=share) 的实例。vcpu_pin_set 将用于提供 VCPU 清单, 并确定固定和未固定实例可以调度到的主机 CPU。 <p>当删除 vcpu_pin_set 时, 将在以后的版本中简化此行为。</p> <p>可能的值 :</p> <ul style="list-style-type: none"> ● 以逗号分隔的物理 CPU 号列表, 实例 VCPU 可以从中分配。每个元素应该是单个 CPU 号、一个 CPU 数字范围, 或者后面有一个要排除的 CPU 号。 <p>前一个范围。例如 :</p> <pre>cpu_shared_set = "4-12,^8,15"</pre> <p>相关选项 :</p> <ul style="list-style-type: none"> ● [compute] cpu_dedicated_set : 这是定义应该从中分配 PCPU 资源的位置的对应选项。 ● vcpu_pin_set : 其定义可能会更改此选项行为的传统选项。

配置选项 = 默认值	类型	Description
image_type_exclude_list = []	列表值	<p>此计算节点不支持的镜像格式列表。</p> <p>在某些情况下，可能需要使计算节点拒绝支持昂贵或复杂的镜像格式。此因素由调度程序在使用给定镜像时选择哪些计算节点做出的决策。</p> <p>可能的值：</p> <ul style="list-style-type: none">任何 glance 镜像 disk_format 名称（即 raw、qcow2 等） <p>相关选项：</p> <ul style="list-style-type: none">[scheduler]query_placement_for_image_type_support - 启用根据支持的镜像类型过滤计算，这需要启用才能生效。

配置选项 = 默认值	类型	Description
<p>live_migration_wait_for_vif_plug = True</p>	<p>布尔值</p>	<p>确定源计算主机是否应该等待来自(neutron)网络服务的 network-vif-plugged 事件，然后再将客户机的实际传送到目标计算主机。</p> <p>请注意，此选项在实时迁移的目标主机上读取。如果您在所有计算主机上设置这个选项，则应以通用方式使用相同的网络后端，则不必担心这一点。</p> <p>在开始传输客户机之前，目标计算主机上会进行一些设置，包括插入虚拟接口。根据目标主机上的网络后端，可能会触发 network-vif-plugged 事件，然后在源计算主机接收，源计算可以等待该事件确保在目标主机上设置网络，然后再启动客户机传输。</p> <p>a. note::</p> <div style="border-left: 2px solid black; padding-left: 10px; margin-left: 20px;"> <p>The compute service cannot reliably determine which types of virtual interfaces (<code>port.binding:vif_type`</code>) will send <code>network-vif-plugged`</code> events without an accompanying port <code>binding:host_id`</code> change. Open vSwitch and linuxbridge should be OK, but OpenDaylight is at least one known backend that will not currently work in this case, see bug https://launchpad.net/bugs/1755890 for more details.</p> </div> <p>可能的值：</p> <ul style="list-style-type: none"> ● true : 在启动客户机传输前等待 network-vif-plugged 事件 ● false : 在启动客户机传输前不要等待 network-vif-plugged 事件 (这是旧的行为) <p>相关选项：</p> <ul style="list-style-type: none"> ● [DEFAULT]/vif_plugging_is_fatal: if live_migration_wait_for_vif_plug_timeout 为 True, vif_plugging_timeout 大于 0, 且达到超时, 实时迁移过程会失败, 但客户机传输不会启动到目标主机 ● [DEFAULT]/vif_plugging_timeout: 如果 live_migration_wait_for_vif_plug 为 True, 这会控制超时前等待的时间, 如果 vif_plugging_is_fatal 为 True, 或者只是继续实时迁移

配置选项 = 默认值	类型	Description
max_concurrent_disk_operations = 0	整数值	我们将并行执行的并发磁盘 I-O 密集型操作(glance 镜像下载、镜像格式转换等)如果这设置过高, 则响应时间会受到影响。默认值为 0 表示没有限制。

配置选项 = 默认值	类型	Description
<p>max_disk_devices_to_attach = -1</p>	<p>整数值</p>	<p>允许附加到单个服务器的磁盘设备的最大数量。请注意，服务器支持的磁盘数量取决于所使用的总线。例如，ide 磁盘总线限制为 4 个附加的设备。配置的最大值是在服务器创建、重建、撤离、取消shelve、实时迁移和附加卷期间强制执行的。</p> <p>通常，磁盘总线会自动从设备类型或磁盘设备以及虚拟化类型决定。但是，也可以通过块设备映射或镜像属性指定磁盘总线。有关在块设备映射中指定磁盘总线的更多信息，请参阅 :doc/user/block-device-mapping 中的 disk_bus 字段，以及有关 hw_disk_bus 镜像属性的更多信息，请参阅 https://docs.openstack.org/glance/latest/admin/ueful-image-properties.html。</p> <p>在托管服务器的计算服务上更改 [compute]/max_disk_devices_to_attach 的 Operator 应注意，如果最大量低于已附加到服务器的设备数，则可能会导致重建失败。例如，如果服务器 A 附加了 26 个设备，并且操作员将 [compute]/max_disk_devices_to_attach 改为 20，则重建服务器 A 的请求将失败，并进入 ERROR 状态，因为 26 设备已连接并超过新配置的最大值 20。</p> <p>Operator 设置 [compute]/max_disk_devices_to_attach 也请注意，在冷迁移过程中，配置的最大值仅会被强制使用，在移动前不会检查目的地。这意味着，如果运维人员在计算主机 A 上设置了最多 26 个，在计算主机 B 上设置了最多 20 个，则从计算主机 A 中的一个附加了 26 个设备的主机冷迁移到计算主机 B 将成功。然后，当服务器位于计算主机 B 上后，重建服务器的后续请求将失败，并进入 ERROR 状态，因为 26 个设备已附加并超过计算主机 B 上配置的最大 20 个。</p> <p>配置的最大值不会在 shelved 卸载服务器上强制使用，因为它们没有计算主机。</p> <p>a. 警告：如果此选项设为 0，nova-compute 服务将无法启动，因为 0 磁盘设备是一个无效的配置，导致实例无法启动。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● -1 表示无限 ● 任何整数 ≥ 1 代表允许的最大值。值 0 将导致 nova-compute 服务无法启动，因为 0 磁盘设备是一个无效的配置，导致实例无法引导。

配置选项 = 默认值	类型	Description
provider_config_location = <code>/etc/nova/provider_config</code> /	字符串值	<p>包含资源供应商配置数据的 YAML 文件的位置。</p> <p>这些文件允许操作员指定额外的自定义清单和特征来分配给一个或多个资源提供程序。</p> <p>有关其他文档，请访问：</p> <p>https://docs.openstack.org/nova/latest/admin/managing-resource-providers.html</p>
resource_provider_association_refresh = 300	整数值	<p>更新计算节点资源提供程序清单、聚合和特征的 nova-compute-side 缓存的时间间隔。</p> <p>此选项指定尝试更新提供程序清单、聚合和计算节点的本地缓存中的秒数。</p> <p>值为零可完全禁用缓存刷新。</p> <p>通过将 SIGHUP 发送到计算进程，可以随时手动清除缓存，从而导致在下一次访问数据时重新填充缓存。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何正整数（以秒为单位），或零来禁用刷新。
shutdown_retry_interval = 10	整数值	<p>在向实例重新发送 ACPI 关闭信号前等待的时间（以秒为单位）。</p> <p>等待的总时间由 shutdown_timeout 设置。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何大于 0 的整数（以秒为单位） <p>相关选项：</p> <ul style="list-style-type: none"> shutdown_timeout
vmdk_allowed_types = ['streamOptimized', 'monolithicSparse']	列表值	<p>描述允许的 VMDK "create-type" 子格式的字符串列表。建议仅包含 single-file-with-sparse-header 变体，以避免因为处理命名扩展而造成潜在的主机文件暴露。如果此列表为空，则不允许使用 VMDK 镜像形式。</p>

9.1.8. conductor

下表概述了 `/etc/nova/nova.conf` 文件中的 **[conductor]** 组下可用的选项。

表 9.7. conductor

配置选项 = 默认值	类型	Description
Worker = None	整数值	OpenStack Conductor 服务的 worker 数量。默认值为可用的 CPU 数量。

9.1.9. 控制台

下表概述了 `/etc/nova/nova.conf` 文件中的 `[console]` 组下可用的选项。

表 9.8. 控制台

配置选项 = 默认值	类型	Description
allowed_origins = []	列表值	<p>在控制台 websocket 代理中添加允许的源列表，以允许来自其他原始主机名的连接。websocket 代理将 host 标头与原始标头匹配，以防止跨站点请求。此列表指定在原始标头中是否允许 host 以外的值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 每个元素都是允许的源主机名的列表，其他一个空列表
ssl_ciphers = None	字符串值	<p>OpenSSL 密码首选项字符串，用于指定允许客户端 TLS 连接的密码。例如：</p> <pre>ssl_ciphers = "kEECDH+aECDH+aAES:kEECDH+aES+aR SA:kEDH+aRSA+aES"</pre> <p>有关 密码 首选项字符串格式和允许的值的详情，请查看 OpenSSL 密码命令的 man page：</p> <pre>https://www.openssl.org/docs/man1.1.0/man1/ ciphers.html</pre> <p>相关选项：</p> <ul style="list-style-type: none"> [DEFAULT] cert [DEFAULT] key
ssl_minimum_version = default	字符串值	<p>允许的 SSL/TLS 协议版本。</p> <p>相关选项：</p> <ul style="list-style-type: none"> [DEFAULT] cert [DEFAULT] key

9.1.10. consoleauth

下表概述了 `/etc/nova/nova.conf` 文件中的 `[consoleauth]` 组下可用的选项。

表 9.9. consoleauth

配置选项 = 默认值	类型	Description
<code>token_ttl = 600</code>	整数值	控制台身份验证令牌的生命周期（以秒为单位）。 在授权用户的控制台访问中使用控制台身份验证令牌。在验证令牌时间过后，令牌将被视为过期。然后，已删除已过期的令牌。

9.1.11. CORS

下表概述了 `/etc/nova/nova.conf` 文件中的 `[cors]` 组下可用的选项。

表 9.10. CORS

配置选项 = 默认值	类型	Description
<code>allow_credentials = True</code>	布尔值	指明实际请求可以包含用户凭证
<code>allow_headers = ['X-Auth-Token', 'X-Openstack-Request-Id', 'X-Identity-Status', 'X-Roles', 'X-Service-Catalog', 'X-User-Id', 'X-Tenant-Id', 'X-OpenStack-Nova-API-Version', 'OpenStack-API-Version']</code>	列表值	指定在实际请求过程中可以使用哪些标头字段名称。
<code>allow_methods = ['GET', 'PUT', 'POST', 'DELETE', 'PATCH']</code>	列表值	指明在实际请求期间可以使用哪些方法。
<code>allowed_origin = None</code>	列表值	指明此资源是否可以与请求"origin"标头中收到的域共享。格式： <code>"<protocol>://<host>[:<port>]"</code> ，没有尾部斜杠。示例： https://horizon.example.com
<code>expose_headers = ['X-Auth-Token', 'X-Openstack-Request-Id', 'X-Subject-Token', 'X-Service-Token', 'X-OpenStack-Nova-API-Version', 'OpenStack-API-Version']</code>	列表值	指明哪些标头可以安全地公开给 API。默认为 HTTP Simple Headers。
<code>max_age = 3600</code>	整数值	CORS preflight 请求的最大缓存期限。

9.1.12. cyborg

下表概述了 `/etc/nova/nova.conf` 文件中的 `[cyborg]` 组下可用的选项。

表 9.11. cyborg

配置选项 = 默认值	类型	Description
<code>cafile = None</code>	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
<code>certFile = None</code>	字符串值	PEM 编码的客户端证书文件
<code>collect-timing = False</code>	布尔值	收集每个 API 调用计时信息。
<code>connect-retries = None</code>	整数值	应该尝试连接错误的最大重试次数。
<code>connect-retry-delay = None</code>	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
<code>endpoint-override = None</code>	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 <code>version</code> , <code>min-version</code> , 和/或 <code>max-version</code> 选项。
<code>insecure = False</code>	布尔值	验证 HTTPS 连接。
<code>keyfile = None</code>	字符串值	PEM 编码的客户端证书密钥文件
<code>region-name = None</code>	字符串值	用于端点 URL 发现的默认 <code>region_name</code> 。
<code>service-name = None</code>	字符串值	用于端点 URL 发现的默认 <code>service_name</code> 。
<code>service-type = accelerator</code>	字符串值	端点 URL 发现的默认 <code>service_type</code> 。
<code>split-loggers = False</code>	布尔值	将请求记录到多个日志记录器。
<code>status-code-retries = None</code>	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
<code>status-code-retry-delay = None</code>	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
<code>timeout = None</code>	整数值	http 请求的超时值
<code>valid-interfaces = ['internal', 'public']</code>	列表值	首选接口列表，用于端点 URL。

9.1.13. database

下表概述了 `/etc/nova/nova.conf` 文件中的 `[database]` 组下可用的选项。

表 9.12. database

配置选项 = 默认值	类型	Description
backend = sqlalchemy	字符串值	用于数据库的后端。
connection = None	字符串值	用于连接到数据库的 SQLAlchemy 连接字符串。
connection_debug = 0	整数值	SQL 调试信息的详细程度：0=None, 100=Everything。
<code>connection_parameters = `</code>	字符串值	用于连接时附加到连接 URL 的可选 URL 参数；指定 <code>param1=value1&param2=value2&...</code>
connection_recycle_time = 3600	整数值	连接池中存在的连接的时间超过这个秒数，下一次从池中签出它们的时间将替换为新的连接。
connection_trace = False	布尔值	将 Python 堆栈跟踪作为注释字符串添加到 SQL。
db_inc_retry_interval = True	布尔值	如果为 True，将数据库操作的重试间隔增加到 <code>db_max_retry_interval</code> 。
db_max_retries = 20	整数值	在引发错误前出现连接错误或死锁错误时的最大重试次数。设置为 -1 以指定无限重试计数。
db_max_retry_interval = 10	整数值	如果设置了 <code>db_inc_retry_interval</code> ，则数据库操作重试之间的最大秒数。
db_retry_interval = 1	整数值	数据库事务重试之间的秒数。
max_overflow = 50	整数值	如果设置，请将这个值用于 SQLAlchemy 的 <code>max_overflow</code> 。
max_pool_size = 5	整数值	在池中保留的最大 SQL 连接数。将值设为 0 表示没有限制。
max_retries = 10	整数值	启动期间数据库连接重试的最大数量。设置为 -1 以指定无限重试计数。
mysql_enable_ndb = False	布尔值	如果为 True，则透明地启用对处理 MySQL 集群 (NDB) 的支持。
mysql_sql_mode = TRADITIONAL	字符串值	用于 MySQL 会话的 SQL 模式。这个选项（包括默认选项）会覆盖任何 <code>server-set</code> SQL 模式。要使用服务器配置设置的任何 SQL 模式，请将其设置为 <code>no</code> 值。示例： <code>mysql_sql_mode=</code>

配置选项 = 默认值	类型	Description
pool_timeout = None	整数值	如果设置，请将这个值用于 SQLAlchemy 的 pool_timeout。
retry_interval = 10	整数值	重新尝试打开 SQL 连接之间的间隔。
slave_connection = None	字符串值	用于连接到从数据库的 SQLAlchemy 连接字符串。
sqlite_synchronous = True	布尔值	如果为 True，则 SQLite 使用同步模式。
use_db_reconnect = False	布尔值	启用在连接丢失时进行数据库重新连接的实验性使用。
use_tpool = False	布尔值	启用所有 DB API 调用的线程池的实验性使用

9.1.14. devices

下表概述了 `/etc/nova/nova.conf` 文件中的 `[devices]` 组下可用的选项。

表 9.13. devices

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
enabled_vgpu_types = []	列表值	<p>计算节点上启用 vGPU 类型。</p> <p>有些 pGPU（如 NVIDIA GRID K1）支持不同的 vGPU 类型。用户可以使用此选项指定可分配给客户机实例的启用 vGPU 类型的列表。</p> <p>如果提供多个 vGPU 类型，则必须将每个 vGPU 类型 额外部分 [vgpu_\$(VGPU_TYPE)] 添加到配置文件中。然后，每个部分 都必须使用 单一配置选项 device_addresses 配置，这是分配给此类型的 PCI 地址列表。</p> <p>如果缺少一个或多个部分（因为特定类型并不想用于至少一个物理 GPU），或者未提供设备地址，则 Nova 将只使用由 [devices]/enabled_vgpu_types 提供的第一个类型。</p> <p>如果为两种不同的类型提供了相同的 PCI 地址，nova-compute 会在重启时返回 InvalidLibvirtGPUConfig 异常。</p> <p>示例如下</p> <pre>[devices] enabled_vgpu_types = nvidia-35, nvidia-36</pre> <pre>[vgpu_nvidia-35] device_addresses = 0000:84:00.0,0000:85:00.0</pre> <pre>[vgpu_nvidia-36] device_addresses = 0000:86:00.0</pre>

9.1.15. ephemeral_storage_encryption

下表概述了 `/etc/nova/nova.conf` 文件中的 **[ephemeral_storage_encryption]** 组下可用的选项。

表 9.14. ephemeral_storage_encryption

配置选项 = 默认值	类型	Description
cipher = aes-xts-plain64	字符串值	<p>要使用的 cipher-mode 字符串。</p> <p>用于加密临时存储的密码和模式。可用的密码模式组合取决于内核支持。根据 dm-crypt 文档，密码应该采用的格式：<code>"<cipher>-<chainmode>-<ivmode>"</code>。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何在 <code>/proc/crypto</code> 中列出的加密选项。

配置选项 = 默认值	类型	Description
enabled = False	布尔值	启用/禁用 LVM 临时存储加密。
key_size = 512	整数值	加密密钥长度（以位为单位）。 用于加密临时存储的加密密钥的位长度。在 XTS 模式中，只有一半的位用于加密密钥。

9.1.16. filter_scheduler

下表概述了 `/etc/nova/nova.conf` 文件中的 `[filter_scheduler]` 组下可用的选项。

表 9.15. filter_scheduler

配置选项 = 默认值	类型	Description
aggregate_image_properties_isolation_namespace = None	字符串值	<p>用于主机聚合的镜像属性命名空间。</p> <p>可以配置镜像和主机，以便某些镜像只能调度到特定聚合中的主机。这使用通过以此选项值开头的主机聚合上设置的元数据值完成。如果主机是带有此类元数据键的聚合的一部分，则请求 spec 中的镜像必须具有其属性中该元数据的值，以便调度程序将主机视为可以接受。</p> <p>请注意，此设置仅在启用了 AggregateImagePropertiesIsolation 过滤器时影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 字符串，其中字符串对应于镜像属性命名空间 <p>相关选项：</p> <ul style="list-style-type: none"> [filter_scheduler] aggregate_image_properties_isolation_separator

配置选项 = 默认值	类型	Description
aggregate_image_properties_isolation_separator = .	字符串值	<p>镜像属性命名空间和名称的分隔符字符。</p> <p>使用 <code>aggregate_image_properties_isolation</code> 过滤器时，相关的元数据键使用 <code>aggregate_image_properties_isolation_namespace</code> 配置选项中定义的命名空间作为前缀，加上分隔符。此选项定义要使用的分隔符。</p> <p>请注意，此设置仅在启用了 AggregateImagePropertiesIsolation 过滤器时影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 字符串，其中字符串对应于镜像属性命名空间分隔符字符 <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] aggregate_image_properties_isolation_namespace
available_filters = ['nova.scheduler.filters.all_filters']	多值	<p>调度程序可以使用的过滤器。</p> <p>nova 调度程序可能应用的过滤器类的未排序列表。只有 [filter_scheduler] enabled_filters 选项中指定的过滤器才会被使用，但所有出现在该选项的过滤器还必须包含在此列表中。</p> <p>默认情况下，这设置为 nova 中包含的所有过滤器。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个零个或多个字符串列表，每个字符串对应于可用于选择主机的过滤器的名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] enabled_filters

配置选项 = 默认值	类型	Description
build_failure_weight_multiplier = 1000000.0	浮点值	<p>用于权衡最近构建失败的主机的倍数。</p> <p>此选项决定在具有最新构建失败的计算节点上放置多少权重。构建失败可能表示故障、配置错误或处于不当的计算节点，并避免在调度期间出现它很有用。权重与计算节点所经历的最新构建失败数量成比例。这个值应该设置为一些高的值，因为可用资源而被其他启用的 <code>weighers</code> 给出的偏移权重。要通过最近失败次数禁用计算主机，请将其设置为零。</p> <p>请注意，此设置仅在启用了 BuildFailureWeigher <code>weigher</code> 时影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 整数或浮点值，其中值对应于此 <code>weigher</code> 的倍数比率。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [compute] consecutive_build_service_disable_threshold - 必须为零，以便计算报告此 <code>weigher</code> 所考虑的数据。 ● [filter_scheduler] weight_classes
cpu_weight_multiplier = 1.0	浮点值	<p>CPU 权重倍数比率。</p> <p>用于权重空闲 vCPU 的倍数。负数表示堆栈而非分散。</p> <p>请注意，只有启用了 CPUWeigher <code>weigher</code> 时，此设置才会影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 整数或浮点值，其中值对应于此 <code>weigher</code> 的倍数比率。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] weight_classes

配置选项 = 默认值	类型	Description
<p>cross_cell_move_weight_multiplier = 1000000.0</p>	<p>浮点值</p>	<p>在跨移动期间用于权衡主机的倍数。</p> <p>此选项决定了在主机上放置了在移动服务器时在同一源单元中的权重数量，例如在跨单元调整大小的主机上。默认情况下，在移动实例时，调度程序将首选同一单元格内的主机，因为跨迁移复杂性质的跨移动操作可能较慢和风险。</p> <p>请注意，只有启用了 CrossCellWeigher weigher 时，此设置才会影响调度。如果您的云没有配置为支持跨迁移，则这个选项无效。</p> <p>此配置选项的值可以通过设置相同名称的聚合元数据键(cross_cell_move_weight_multiplier)来覆盖每个主机聚合。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 整数或浮点值，其中值对应于此 weigher 的倍数比率。正值表示，eigher 将优先选择实例当前运行的同一单元中的主机。负值意味着，在实例当前运行的 其他 单元中首选主机。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] weight_classes
<p>disk_weight_multiplier = 1.0</p>	<p>浮点值</p>	<p>磁盘权重多器比率。</p> <p>用于权衡可用磁盘空间的倍数。负数意味着堆栈与分布。</p> <p>请注意，只有启用了 DiskWeigher weigher 时，此设置才会影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 整数或浮点值，其中值对应于此 weigher 的多条比率。

配置选项 = 默认值	类型	Description
<pre>enabled_filters = ['AvailabilityZoneFilter', 'ComputeFilter', 'ComputeCapabilitiesFilter', 'ImagePropertiesFilter', 'ServerGroupAntiAffinityFilter', 'ServerGroupAffinityFilter']</pre>	列表值	<p>调度程序将使用的过滤器。</p> <p>用于过滤主机的过滤类名称的有序列表。这些过滤器将按照列出的顺序应用，以便首先将限制性最严格的过滤器放在最严格的过滤器中，以便提高过滤过程的效率。</p> <p>此选项中的所有过滤器 都必须 出现在 [scheduler_filter] available_filter 选项中，否则将引发 SchedulerHostFilterNotFound 异常。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 一个零个或多个字符串列表，每个字符串对应于要用于选择主机的过滤器的名称 <p>相关选项：</p> <ul style="list-style-type: none"> [filter_scheduler] available_filters
<pre>host_subset_size = 1</pre>	整数值	<p>调度程序选择的最佳主机子集的大小。</p> <p>新实例将被调度到从 N 最佳主机的子集随机选择的主机上，其中 N 是此选项设置的值。</p> <p>把它设置为大于 1 的值将减少多个调度程序处理类似请求的几率将选择同一主机，从而造成潜在的竞争条件。从最适合请求的 N 主机中随机选择主机，从而降低冲突的可能性。但是，您设置这个值越高，所选主机对于给定请求可能的最佳选择。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 一个整数，整数对应于主机子集的大小。

配置选项 = 默认值	类型	Description
<p>hypervisor_version_weight_multiplier = 1.0</p>	<p>浮点值</p>	<p>虚拟机监控程序版本权重比率。</p> <p>倍数用于根据报告的 hypervisor 版本对主机进行权重。负数表示首选使用较旧的主机，默认为首选较新的主机来协助升级。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 整数或浮点值，其中值对应于此 weigher 的倍数比率。 <p>Example:</p> <ul style="list-style-type: none"> ● 强烈建议使用旧的主机 <ul style="list-style-type: none"> a. code-block:: ini <pre data-bbox="970 757 1447 891">[filter_scheduler] hypervisor_version_weight_multiplier=-1000</pre> ● 中等首选新主机 <ul style="list-style-type: none"> a. code-block:: ini <pre data-bbox="970 1032 1447 1167">[filter_scheduler] hypervisor_version_weight_multiplier=2.5</pre> ● 禁用 weigher influence <ul style="list-style-type: none"> a. code-block:: ini <pre data-bbox="970 1308 1447 1442">[filter_scheduler] hypervisor_version_weight_multiplier=0</pre> <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] weight_classes
<p>image_properties_default_architecture = None</p>	<p>字符串值</p>	<p>使用镜像属性过滤器时要使用的默认架构。</p> <p>使用 ImagePropertiesFilter 时，您可以定义一个默认架构来方便用户体验，并避免像 x86_64 镜像在 AARCH64 计算节点上登录的内容，因为用户没有在 Glance 中指定 hw_architecture 属性。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● CPU 架构，如 x86_64、aarch64、s390x。

配置选项 = 默认值	类型	Description
io_ops_weight_multiplier = -1.0	浮点值	<p>IO 操作权重多器比率。</p> <p>这个选项决定了具有不同工作负载的主机是如何权衡的。负值（如默认值）将导致调度程序首选使用更轻量的工作负载的主机，而正值将首选具有高度工作负载的主机。查看它的另一个方法是此选项的正值会将实例调度到已经忙碌的主机上，而负值则往往在更多主机之间分发工作负载。绝对值（无论是正还是负）控制 io_ops weigher 相对于其他 Weighers 的强度量。</p> <p>请注意，只有启用了 IoOpsWeigher weigher 时，此设置才会影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 整数或浮点值，其中值对应于此 weigher 的多条比率。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] weight_classes
isolated_hosts = []	列表值	<p>只能运行某些镜像的主机列表。</p> <p>如果需要将一些镜像限制为仅在特定的指定主机上运行，请在此处列出这些主机名。</p> <p>请注意，此设置仅在启用了 IsolatedHostsFilter 过滤器时影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 字符串列表，每个字符串对应于主机的名称 <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] isolated_images ● [filter_scheduler] restrict_isolated_hosts_to_isolated_images

配置选项 = 默认值	类型	Description
isolated_images = []	列表值	<p>只能在特定主机上运行的镜像的 UUID 列表。</p> <p>如果要将某些镜像限制为仅在特定指定的主机上运行，请在此处列出这些镜像 UUID。</p> <p>请注意，此设置仅在启用了 IsolatedHostsFilter 过滤器时影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • UUID 字符串列表，其中每个字符串都对应于镜像的 UUID <p>相关选项：</p> <ul style="list-style-type: none"> • [filter_scheduler] isolated_hosts • [filter_scheduler] restrict_isolated_hosts_to_isolated_images
max_instances_per_host = 50	整数值	<p>主机上可以存在的最大实例数。</p> <p>如果您需要限制任何给定主机上的实例数量，请将此选项设置为您要允许的最大实例数量。</p> <p>NumInstancesFilter 和 AggregateNumInstancesFilter 将拒绝任何至少具有多于此选项值的实例的主机。</p> <p>请注意，此设置仅在启用了 NumInstancesFilter 或 AggregateNumInstancesFilter 过滤器时影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • 整数，其中整数对应于主机上可调度的最大实例。 <p>相关选项：</p> <ul style="list-style-type: none"> • [filter_scheduler] enabled_filters

配置选项 = 默认值	类型	Description
max_io_ops_per_host = 8	整数值	<p>在主机上主动执行 IO 的实例数量。</p> <p>执行 IO 的实例包括以下状态：build, resize, snapshot, migrate, rescue, unshelve。</p> <p>请注意，此设置仅在启用了 IoOpsFilter 过滤器时影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 整数，其中整数对应于任意给定主机上可以主动执行 IO 的最大实例数。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] enabled_filters
pci_weight_multiplier = 1.0	浮点值	<p>PCI 设备关联性权重倍数。</p> <p>PCI 设备关联性权重根据主机上的 PCI 设备数量以及实例请求的 PCI 设备数量计算权重。</p> <p>请注意，只有启用了 PCIWeigher weigher 和 NUMATopologyFilter 过滤器时，此设置才会影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数或浮点值，其中值对应于此 weigher 的倍数比率。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] weight_classes

配置选项 = 默认值	类型	Description
ram_weight_multiplier = 1.0	浮点值	<p>RAM 权重多器比率。</p> <p>这个选项决定了具有更多或小于可用 RAM 的主机如何权衡。正值将导致调度程序首选使用更多可用 RAM 的主机，负数会导致调度程序首选使用较少的可用主机。查看它的另一个方法是此选项的正值往往会将实例分散到多个主机上，而负值往往会在调度到较早的主机前填充(stack)主机。绝对值（无论是正还是负值）控制 RAM weigher 相对于其他 Weighers 的强度量。</p> <p>请注意，只有启用了 RAMWeigher weigher 时，此设置才会影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 整数或浮点值，其中值对应于此 weigher 的多条比率。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] weight_classes
restrict_isolated_hosts_to_isolated_images = True	布尔值	<p>防止在隔离主机上构建非隔离镜像。</p> <p>请注意，此设置仅在启用了 IsolatedHostsFilter 过滤器时影响调度。即使如此，这个选项不会影响对隔离镜像的请求行为，这将 始终 限制为隔离的主机。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] isolated_images ● [filter_scheduler] isolated_hosts
shuffle_best_same_weighted_hosts = False	布尔值	<p>启用在具有同样最佳权重的主机间分散实例。</p> <p>当 [filter_scheduler] host_subset_size 为 1（默认），但有大量的主机具有相同的 maximal weight 时，启用它很有用。在 Ironic 部署中，这种场景很常见，其中通常有很多裸机节点返回了相同的权重。在这种情况下，启用这个选项会减少重新调度事件的争用和机会。同时，它将使实例打包（即使在未权衡的情况下）会减少密度。</p>

配置选项 = 默认值	类型	Description
soft_affinity_weight_multiplier = 1.0	浮点值	<p>用于权衡主机用于组软关联性的倍数。</p> <p>请注意，只有启用了 ServerGroupSoftAffinityWeigher weigher 时，此设置才会影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 非负整数或浮点值，其中值对应于具有组软关联性的主机的权重倍数。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] weight_classes
soft_anti_affinity_weight_multiplier = 1.0	浮点值	<p>用于 weighing 主机用于组 soft-anti-affinity 的倍数。</p> <p>请注意，只有启用了 ServerGroupSoftAntiAffinityWeigher weigher 时，此设置才会影响调度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 非负整数或浮点值，其中值对应于具有组软反关联性的主机的权重倍数。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] weight_classes

配置选项 = 默认值	类型	Description
track_instance_changes = True	布尔值	<p>启用查询单个主机以获取实例信息。</p> <p>调度程序可能需要有关主机上实例的信息，才能评估其过滤器和权重。此信息的最常见需要是(anti-)affinity 过滤器，它需要根据已在主机上运行的实例选择主机。</p> <p>如果配置的过滤器和权重不需要此信息，禁用这个选项将提高性能。当跟踪开销证明太重时，也可以禁用它，但这会导致需要主机使用数据在每个请求上查询数据库的类。</p> <p>a. note::</p> <div style="border-left: 2px solid black; padding-left: 10px; margin-left: 20px;"> <p>In a multi-cell (v2) setup where the cell MQ is separated from the top-level, computes cannot directly communicate with the scheduler. Thus, this option cannot be enabled in that scenario. See also the <code>[workarounds] disable_group_policy_check_upcall</code> option.</p> </div> <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] enabled_filters ● [workarounds] disable_group_policy_check_upcall
weight_classes = ['nova.scheduler.weights.all_weighers']	列表值	<p>调度程序将使用的权衡。</p> <p>只有通过过滤器的主机才会权衡。任何主机的权重从 0 开始，而 <code>weighers</code> 通过添加到或从之前所分配给的权重来对这些主机进行排序。权重可能会成为负数。实例将调度到 N 个最轻量级主机之一，其中 N 是 [filter_scheduler] host_subset_size。</p> <p>默认情况下，这设置为 Nova 中包含的所有 <code>weighers</code>。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 一个零个或多个字符串列表，每个字符串都对应于用于选择主机的 <code>weigher</code> 的名称

9.1.17. Glance

下表概述了 `/etc/nova/nova.conf` 文件中的 **[glance]** 组下可用的选项。

表 9.16. Glance

配置选项 = 默认值	类型	描述
api_servers = None	列表值	<p>nova 可用的 glance api 服务器端点列表。</p> <p>HTTPS 用于基于 ssl 的 glance api 服务器。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>端点发现的首选机制是通过 keystoneauth1 加载选项。只有您需要多个端点且因为某种原因无法使用负载均衡器时，才使用 api_servers。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> 任何以 "scheme://hostname:port[/path]" (如 "http://10.0.1.0:9292" 或 "https://my.glance.server/image") 格式的完全限定域名。 <p>弃用自： 21.0.0</p> <p>reason： 在 17.0.0 Queens 发行版本中添加了标准 keystoneauth1 Adapter 选项支持镜像服务配置。api_servers 选项被临时保留，以便消费者能够缩短实时负载均衡解决方案。</p>
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
connect-retries = None	整数值	应该尝试连接错误的最大重试次数。
connect-retry-delay = None	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
debug = False	布尔值	使用 glanceclient 启用或禁用调试日志记录。

配置选项 = 默认值	类型	描述
default_trusted_certificate_ids = []	列表值	<p>应该信任的证书的证书 ID 列表。</p> <p>可作用于证书验证的可信证书 ID 的默认列表。如果用户提供带有实例 API 请求的可信证书 ID 列表，则这个选项的值将被忽略。如果启用了签名验证和证书验证，且用户未提供替代列表，则此选项的值将保留给实例数据。如果启用了证书验证，用户必须提供可信证书 ID 列表，否则证书验证将失败。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● 如果启用了 <code>verify_glance_signatures</code> 和 <code>enable_certificate_validation</code>，则可以使用这个选项的值。
enable_certificate_validation = False	布尔值	<p>为镜像签名验证启用证书验证。</p> <p>在镜像签名验证 nova 过程中，首先使用与实例关联的可信证书集合来验证镜像的签名证书的有效性。如果证书验证失败，则不会执行签名验证，实例将放入错误状态。这为最终用户提供了强大的保证，确保镜像数据未经修改且值得信任。如果禁用了镜像签名，则仍然可能会出现镜像签名验证，但最终用户不会有任何保证用于生成镜像签名的签名证书仍然值得信任。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● 这个选项仅在启用 <code>verify_glance_signatures</code> 时才会生效。 ● 启用此选项时，可以使用 <code>default_trusted_certificate_ids</code> 的值。 <p>弃用自： 16.0.0</p> <p>原因： 此选项旨在简化使用镜像签名验证部署的转换。预期的状态长期用于签名验证和证书验证，以始终一起进行。</p>

配置选项 = 默认值	类型	描述
enable_rbd_download = False	布尔值	<p>启用直接通过 RBD 下载 Glance 镜像。</p> <p>允许计算主机直接从 Ceph 下载和缓存镜像，而不是从 Glance API 中下载并缓存操作速度。这可能会使下载几十到几百 GB 镜像所需的时间从几十分钟减少到及十秒，但这需要基于 Ceph 的部署并需要可以从计算节点访问到 Ceph。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● [glance] rbd_user ● [glance] rbd_connect_timeout ● [glance] rbd_pool ● [glance] rbd_ceph_conf
endpoint-override = None	字符串值	<p>始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version, min-version, 和/或 max-version 选项。</p>
insecure = False	布尔值	<p>验证 HTTPS 连接。</p>
keyfile = None	字符串值	<p>PEM 编码的客户端证书密钥文件</p>
num_retries = 3	整数值	<p>启用 glance 操作重试。</p> <p>指定从 glance 上传 / 将镜像下载到 / 时的重试次数。0 表示没有重试。</p>
<code>`rbd_ceph_conf = `</code>	字符串值	<p>要使用的 ceph 配置文件的路径。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● 只有在 [glance] enable_rbd_download 设为 True 时才会使用此选项。
rbd_connect_timeout = 5	整数值	<p>最初连接到集群时 RADOS 客户端超时（以秒为单位）。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● 只有在 [glance] enable_rbd_download 设为 True 时才会使用此选项。

配置选项 = 默认值	类型	描述
<code>`rbd_pool = `</code>	字符串值	Glance 镜像存储为 rbd 卷的 RADOS 池。 相关选项： <ul style="list-style-type: none">只有在 [glance] enable_rbd_download 设为 True 时才会使用此选项。
<code>`rbd_user = `</code>	字符串值	用于访问作为 rbd 卷存储的 Glance 镜像的 RADOS 客户端名称。 相关选项： <ul style="list-style-type: none">只有在 [glance] enable_rbd_download 设为 True 时才会使用此选项。
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = image	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
timeout = None	整数值	http 请求的超时值
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。

配置选项 = 默认值	类型	描述
verify_glance_signatures = False	布尔值	<p>启用镜像签名验证。</p> <p>Nova 使用来自 glance 的镜像签名元数据，并在下载该镜像时验证签名镜像的签名。如果无法验证镜像签名，或者镜像签名元数据不完整或不可用，则 nova 不会引导镜像，而是会将实例置于错误状态。这为最终用户提供了创建服务器的镜像数据的完整性更强的保证。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● key_manager 组中的选项，因为 key_manager 用于签名验证。 ● 以下 enable_certificate_validation 和 default_trusted_certificate_ids 均依赖于正在启用此选项。

9.1.18. guestfs

下表概述了 `/etc/nova/nova.conf` 文件中的 `[guestfs]` 组下可用的选项。

表 9.17. guestfs

配置选项 = 默认值	类型	Description
debug = False	布尔值	<p>启用/禁用 guestfs 日志记录。</p> <p>这将配置 guestfs 以调试消息并将其推送到 OpenStack 日志记录系统。当设置为 True 时，它会跟踪 libguestfs API 调用并启用详细的调试消息。要使用上述功能，必须安装"libguestfs"软件包。</p> <p>相关选项：</p> <p>由于 libguestfs 访问权限和修改由 libvirt 管理的虚拟机，因此应设置以下选项以授予这些虚拟机的访问权限。</p> <ul style="list-style-type: none"> ● libvirt.inject_key ● libvirt.inject_partition ● libvirt.inject_password

9.1.19. healthcheck

下表概述了 `/etc/nova/nova.conf` 文件中的 `[healthcheck]` 组下可用的选项。

表 9.18. healthcheck

配置选项 = 默认值	类型	Description
backends = []	列表值	可执行健康检查并报告该信息作为请求的一部分的其他后端。
detailed = False	布尔值	显示响应中更详细的信息。安全备注：启用此选项可能会公开有关被监控服务的敏感详情。务必验证它是否不会违反您的安全策略。
disable_by_file_path = None	字符串值	检查是否存在文件，以确定应用是否在端口上运行。DisableByFileHealthcheck 插件使用。
disable_by_file_paths = []	列表值	根据端口检查文件是否存在，以确定应用是否在端口上运行。需要字符串的 "port:path" 列表。DisableByFilesPortsHealthcheck 插件使用。
path = /healthcheck	字符串值	响应 healthcheck 请求的路径。

9.1.20. hyperv

下表概述了 `/etc/nova/nova.conf` 文件中的 `[hyperv]` 组下可用的选项。

表 9.19. hyperv

配置选项 = 默认值	类型	Description
config_drive_cdrom = False	布尔值	<p>将配置驱动器挂载为 CD 驱动器。</p> <p>OpenStack 可以配置为将实例元数据写入配置驱动器，然后附加到实例，然后再启动。配置驱动器可以附加为磁盘驱动器（默认）或 CD 驱动器。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● 这个选项对 <code>force_config_drive</code> 选项设为 <code>True</code> 或创建实例的 REST API 调用将具有 <code>--config-drive=True</code> 标志。 ● <code>config_drive_format</code> 选项必须设置为 <code>iso9660</code>，以便将 CD 驱动器用作配置驱动器镜像。 ● 要将配置驱动器与 Hyper-V 一起使用，您必须将 <code>mkisofs_cmd</code> 值设置为 <code>mkisofs.exe</code> 安装的完整路径。另外，您必须将 <code>qemu_img_cmd</code> 值设置为到 <code>qemu-img</code> 命令安装的完整路径。 ● 您可以通过将 <code>force_config_drive</code> 选项设置为 <code>True</code>，将 Compute 服务配置为始终创建配置驱动器。

配置选项 = 默认值	类型	Description
config_drive_inject_password = False	布尔值	<p>将密码注入配置驱动器。</p> <p>启用后，可以从配置驱动器镜像获取 admin 密码。</p> <p>相关选项：</p> <ul style="list-style-type: none"> 这个选项与启用 Hyper-V 的 config 驱动器使用情况的其他选项（如 force_config_drive）一起使用时有意义。
dynamic_memory_ratio = 1.0	浮点值	<p>动态内存比率</p> <p>当设置为大于 1 的值时，启用动态内存分配 (ballooning)。该值表示分配给实例的 RAM 总量及其启动 RAM 量之间的比率。例如，对于具有 1024MB RAM 的实例，比例为 2.0，代表启动时分配的 512MB RAM。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 1.0：禁用动态内存分配（默认）。 大于 1.0 的浮点值：启用分配总表示的 RAM 的分配，以便启动这个值。
enable_instance_metrics_collection = False	布尔值	<p>启用实例指标集合</p> <p>使用 Hyper-V 的指标 API 为实例启用指标集合。收集的数据可由其他应用程序和服务检索，例如：Ceilometer。</p>

配置选项 = 默认值	类型	Description
enable_remotefx = False	布尔值	<p>启用 RemoteFX 功能</p> <p>这至少需要一个适用于 Windows / Hyper-V Server 2012 R2 或更新版本的图形适配器和 RDS 虚拟化功能的 DirectX 11 功能。</p> <p>可以使用以下类型额外规格来请求带有 RemoteFX 的实例：</p> <p>OS : 分辨率.客户虚拟机屏幕分辨率。可接受值 1024x768, 1280x1024, 1600x1200, 1920x1200, 2560x1600, 3840x2160</p> <p>3840x2160 仅在 Windows / Hyper-V Server 2016 中提供。</p> <p>OS : monitor.客户虚拟机监控器数。可接受值 [1, 4] - Windows / Hyper-V Server 2012 R2 [1, 8] - Windows / Hyper-V Server 2016</p> <p>OS:vram.虚拟机 VRAM 数量.仅适用于 Windows / Hyper-V Server 2016。可接受值：</p> <p>64, 128, 256, 512, 1024</p>
<code>`instances_path_share = `</code>	字符串值	<p>实例路径共享</p> <p>映射到 "instances_path" dir 的 Windows 共享名称，并使用调整大小功能将文件复制到目标主机。如果留空，则将使用管理共享（隐藏网络共享），查找本地使用的相同 "instances_path"。</p> <p>可能的值：</p> <ul style="list-style-type: none"> • ""：将使用管理共享（默认）。 • Windows 共享的名称。 <p>相关选项：</p> <ul style="list-style-type: none"> • "instances_path": 此处这个选项为空时将使用的目录。
iscsi_initiator_list = []	列表值	<p>用于稳定化 iSCSI 会话的 iSCSI 启动器列表。</p> <p>如果未指定，则 Microsoft iSCSI 启动器服务将选择启动器。</p>
limit_cpu_features = False	布尔值	<p>限制 CPU 功能</p> <p>需要此标志来支持在实例创建过程中进行实时迁移到具有不同 CPU 功能的主机，以便限制实例使用的 CPU 功能。</p>

配置选项 = 默认值	类型	Description
mounted_disk_query_retry_count = 10	整数值	<p>挂载磁盘查询重试计数</p> <p>重试检查挂载磁盘的次数。查询会运行，直到找到设备或达到重试计数。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数值。建议大于1的值（默认值：10）。 <p>相关选项：</p> <ul style="list-style-type: none"> ● 磁盘挂载重试之间的时间间隔使用 "mounted_disk_query_retry_interval" 选项声明。
mounted_disk_query_retry_interval = 5	整数值	<p>挂载磁盘查询重试间隔</p> <p>检查挂载的磁盘间隔（以秒为单位）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 以秒为单位的时间（默认为5）。 <p>相关选项：</p> <ul style="list-style-type: none"> ● 当 mounted_disk_query_retry_count 大于1时，这个选项是有意义的。 ● 重试循环使用 mounted_disk_query_retry_count 和 mounted_disk_query_retry_interval 配置选项运行。
power_state_check_timeframe = 60	整数值	<p>电源状态检查时间线</p> <p>检查实例电源状态变化的时间线。此选项用于通过 WMI 接口在指定时间段内从 Hyper-V 获取实例状态。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 以秒为单位的时间线（默认值：60）。
power_state_event_polling_interval = 2	整数值	<p>电源状态事件轮询间隔</p> <p>实例电源状态更改事件轮询频率。将电源状态事件的监听程序间隔设置为给定值。这个选项增强了重启自身实例的内部生命周期通知。不太可能更改这个值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 以秒为单位的时间（默认：2）。

配置选项 = 默认值	类型	Description
qemu_img_cmd = qemu-img.exe	字符串值	<p>qemu-img 命令</p> <p>有些镜像相关操作需要 qemu-img，如在不同镜像类型之间进行转换。您可以从 http://qemu.weilnetz.de/ 获得它，或安装 Cloudbase OpenStack Hyper-V Compute Driver (https://cloudbase.it/openstack-hyperv-driver/)，这会自动为这个设置选项设置正确的路径。您可以给出 qemu-img.exe 的完整路径，或者在 PATH 环境变量中设置其路径，并将此选项保留为默认值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● qemu-img 可执行文件的名称，如果它与 nova-compute 服务位于同一个目录中，或者其路径位于 PATH 环境变量（默认）。 ● qemu-img 命令的路径 (DRIVELETTER:\PATH\TO\QEMU-IMG\COMMAND)。 <p>相关选项：</p> <ul style="list-style-type: none"> ● 如果 config_drive_cdrom 选项为 False，则 qemu-img 将用于将 ISO 转换为 VHD，否则配置驱动器将保留 ISO。要将配置驱动器与 Hyper-V 一起使用，您必须将 mkisofs_cmd 值设置为 mkisofs.exe 安装的完整路径。
use_multipath_io = False	布尔值	<p>在附加 iSCSI 或 FC 磁盘时使用多路径连接。</p> <p>这需要启用多路径 IO Windows 功能。MPIO 必须配置为声明这样的设备。</p>
volume_attach_retry_count = 10	整数值	<p>卷附加重试计数</p> <p>重试附加卷的次数。卷附加会重试，直到达到成功或给定重试计数为止。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数值（默认值：10）。 <p>相关选项：</p> <ul style="list-style-type: none"> ● 使用 volume_attach_retry_interval 选项声明附加尝试之间的时间间隔。

配置选项 = 默认值	类型	Description
volume_attach_retry_interval = 5	整数值	<p>卷附加重试间隔</p> <p>卷附加尝试之间的间隔，以秒为单位。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 以秒为单位的时间（默认为 5）。 <p>相关选项：</p> <ul style="list-style-type: none"> 当 <code>volume_attach_retry_count</code> 大于 1 时，这个选项有意义。 重试循环使用 <code>volume_attach_retry_count</code> 和 <code>volume_attach_retry_interval</code> 配置选项运行。
vswitch_name = None	字符串值	<p>外部虚拟交换机名称</p> <p>Hyper-V 虚拟交换机是基于软件的第 2 层以太网网络交换机，可通过安装 Hyper-V 服务器角色获得。交换机包括以编程方式管理且可扩展的功能，可将虚拟机连接到虚拟网络和物理网络。另外，Hyper-V 虚拟交换机为安全、隔离和服务级别提供策略强制。此配置选项表示的 vSwitch 必须是外部的（不是内部或私有）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 如果没有提供，则使用第一个可用 <code>vswitches</code> 列表。此列表通过 WQL 查询。 虚拟交换机名称。
wait_soft_reboot_seconds = 60	整数值	<p>等待软重启秒数</p> <p>在发出软重启请求后等待实例关闭的秒数。如果实例在此窗口中没有关闭，我们回退到硬重启。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 以秒为单位的时间（默认值：60）。

9.1.21. image_cache

下表概述了 `/etc/nova/nova.conf` 文件中的 `[image_cache]` 组下可用的选项。

表 9.20. image_cache

配置选项 = 默认值	类型	描述
manager_interval = 2400	整数值	<p>镜像缓存管理器运行之间等待的秒数。</p> <p>请注意，在多个 nova-compute 服务间为 [DEFAULT]/instances_path 配置选项使用共享存储时，这个定期可能会处理大量实例。同样，使用管理集群（如 vmwareapi.VMwareVCDriver）的计算驱动程序可能会导致处理大量实例。因此，您可能需要调整预期负载的时间间隔，或者仅在共享存储聚合的一个 nova-compute 服务中运行。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0: 以默认间隔 60 秒运行（不推荐） ● -1: disable ● 任何其它值 <p>相关选项：</p> <ul style="list-style-type: none"> ● [DEFAULT]/compute_driver ● [DEFAULT]/instances_path
precache_concurrency = 1	整数值	<p>用于并行触发镜像预缓存的最大计算主机数量。</p> <p>进行镜像预缓存请求时，将联系计算节点以启动下载。这个数字限制了并行发生的数量。数值越大，计算可以并行工作，并可能导致减少时间完成该操作，但也可能是 DDoS 镜像服务。较低数字将导致后续操作，降低镜像服务负载，但可能较长的完成运行时。</p>
remove_unused_base_images = True	布尔值	应该删除未使用的基础镜像？
remove_unused_original_minimum_age_seconds = 86400	整数值	比此年轻而未调整的基础镜像不会被删除。
remove_unused_resized_minimum_age_seconds = 3600	整数值	比此年轻而未使用的基础镜像调整了大小，则不会被删除。
subdirectory_name = _base	字符串值	<p>缓存镜像的位置。</p> <p>这不是完整路径 - 只是相对于 <i>\$instances_path</i> 的文件夹名称。对于 per-compute-host 缓存的镜像，设置为 <i>base\$my_ip</i></p>

9.1.22. ironic

下表概述了 `/etc/nova/nova.conf` 文件中的 **[ironic]** 组下可用的选项。

表 9.21. ironic

配置选项 = 默认值	类型	Description
api_max_retries = 60	整数值	请求冲突时要重试的次数。如果设置为 0，则尝试一次，则不重试。 相关选项： <ul style="list-style-type: none">• api_retry_interval
api_retry_interval = 2	整数值	重试请求前等待的秒数。 相关选项： <ul style="list-style-type: none">• api_max_retries
auth-url = None	字符串值	身份验证 URL
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
connect-retries = None	整数值	应该尝试连接错误的最大重试次数。
connect-retry-delay = None	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
endpoint-override = None	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version , min-version , 和/或 max-version 选项。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件

配置选项 = 默认值	类型	Description
partition_key = None	字符串值	不区分大小写的键，将可能由此服务管理的节点集合限制为 Ironic 中的一组节点，其具有匹配的 conductor_group 属性。如果未设置，则所有可用节点将有资格由此服务管理。请注意，将其设置为空字符串("")将与默认编排器组匹配，并且与保留选项未设置不同。
password = None	字符串值	用户的密码
peer_list = []	列表值	使用这个 partition_key 配置值的所有 nova-compute 服务（包括此主机）的主机名列表。与 partition_key 值匹配的节点将在此处指定的所有服务之间分发。如果未设置 partition_key，这个选项将被忽略。
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
serial_console_state_timeout = 10	整数值	等待节点串口控制台状态更改的超时（秒）。设置为 0 可禁用超时。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = baremetal	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
system-scope = None	字符串值	系统操作的范围
timeout = None	整数值	http 请求的超时值

配置选项 = 默认值	类型	Description
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。

9.1.23. key_manager

下表概述了 `/etc/nova/nova.conf` 文件中的 `[key_manager]` 组下可用的选项。

表 9.22. key_manager

配置选项 = 默认值	类型	Description
auth_type = None	字符串值	要创建的身份验证凭证的类型。可能的值有 <code>令牌</code> 、 <code>密码</code> 、 <code>keystone_token</code> 和 <code>keystone_password</code> 。如果没有将上下文传递给凭证工厂，则需要此项。
auth_url = None	字符串值	使用此端点连接到 Keystone。
backend = barbican	字符串值	指定密钥管理器实现。选项为 "barbican" 和 "vault"。默认为 "barbican"。将支持之前使用 <code>[key_manager]/api_class</code> 来设置的值。
domain_id = None	字符串值	域范围的域 ID。 <code>keystone_token</code> 和 <code>keystone_password</code> <code>auth_type</code> 是可选的。
domain_name = None	字符串值	域范围的域名。 <code>keystone_token</code> 和 <code>keystone_password</code> <code>auth_type</code> 是可选的。
fixed_key = None	字符串值	修复了密钥管理器返回的密钥，在 hex 中指定。 可能的值： <ul style="list-style-type: none"> 空字符串或十六进制值中的键
password = None	字符串值	用于身份验证的密码。 <code>password</code> 和 <code>keystone_password</code> <code>auth_type</code> 需要。

配置选项 = 默认值	类型	Description
project_domain_id = None	字符串值	项目的项目域 ID。 <i>keystone_token</i> 和 <i>keystone_password</i> <i>auth_type</i> 是可选的。
project_domain_name = None	字符串值	项目的项目的域名。 <i>keystone_token</i> 和 <i>keystone_password</i> <i>auth_type</i> 是可选的。
project_id = None	字符串值	项目范围的项目 ID。 <i>keystone_token</i> 和 <i>keystone_password</i> <i>auth_type</i> 是可选的。
project_name = None	字符串值	项目范围的项目名称。 <i>keystone_token</i> 和 <i>keystone_password</i> <i>auth_type</i> 是可选的。
reauthenticate = True	布尔值	如果当前令牌即将过期，则允许获取新令牌。 <i>keystone_token</i> 和 <i>keystone_password</i> <i>auth_type</i> 是可选的。
token = None	字符串值	进行身份验证的令牌。如果没有将上下文传递给凭证工厂， <i>token</i> 和 <i>keystone_token</i> <i>auth_type</i> 需要。
trust_id = None	字符串值	信任范围的信任 ID。 <i>keystone_token</i> 和 <i>keystone_password</i> <i>auth_type</i> 是可选的。
user_domain_id = None	字符串值	用于身份验证的用户域 ID。 <i>keystone_token</i> 和 <i>keystone_password</i> <i>auth_type</i> 是可选的。
user_domain_name = None	字符串值	用于身份验证的用户域名。 <i>keystone_token</i> 和 <i>keystone_password</i> <i>auth_type</i> 是可选的。
user_id = None	字符串值	用于身份验证的用户 ID。 <i>keystone_token</i> 和 <i>keystone_password</i> <i>auth_type</i> 是可选的。
username = None	字符串值	进行身份验证的用户名。 <i>password</i> <i>auth_type</i> 需要。 <i>keystone_password</i> <i>auth_type</i> 是可选的。

9.1.24. Keystone

下表概述了 `/etc/nova/nova.conf` 文件中的 `[keystone]` 组下可用的选项。

表 9.23. Keystone

配置选项 = 默认值	类型	Description
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书文件

配置选项 = 默认值	类型	Description
collect-timing = False	布尔值	收集每个 API 调用计时信息。
connect-retries = None	整数值	应该尝试连接错误的最大重试次数。
connect-retry-delay = None	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
endpoint-override = None	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version , min-version , 和/或 max-version 选项。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = identity	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
timeout = None	整数值	http 请求的超时值
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。

9.1.25. keystone_authtoken

下表概述了 `/etc/nova/nova.conf` 文件中的 `[keystone_authtoken]` 组下可用的选项。

表 9.24. keystone_authtoken

配置选项 = 默认值	类型	Description
auth_section = None	字符串值	从中加载插件特定选项的配置部分

配置选项 = 默认值	类型	Description
auth_type = None	字符串值	要加载的身份验证类型
auth_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。 这个选项已弃用，现在使用 <code>www_authenticate_uri</code>，并将在 S 发行版本中删除。弃用自：Queens *reason:* auth_uri 选项已弃用，现在使用 <code>www_authenticate_uri</code>，并将在 S 发行版本中删除。
auth_version = None	字符串值	Identity API 端点的 API 版本。
cache = None	字符串值	存储 Swift 缓存对象的请求环境密钥。当使用 Swift 缓存部署 <code>auth_token</code> 中间件时，请使用此选项使中间件共享具有 <code>swift</code> 的缓存后端。否则，使用 <code>memcached_servers</code> 选项。
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。默认为系统 CA。
certFile = None	字符串值	如果身份服务器需要客户端证书，则需要此项
delay_auth_decision = False	布尔值	不要在中间件内处理授权请求，而是将授权决策委派给下游 WSGI 组件。
enforce_token_bind = permissive	字符串值	用于控制令牌绑定的使用和类型。可以设置为 "disabled" 以不检查令牌绑定。如果绑定类型是已知服务器的绑定类型，则验证绑定信息，并忽略它（如 "permissive"），但如果绑定类型未知，则令牌将被拒绝。"必需"令牌绑定需要允许任何形式的令牌绑定。最后，令牌中必须存在的绑定方法名称。
http_connect_timeout = None	整数值	请求超时值与身份 API 服务器通信。
http_request_max_retries = 3	整数值	与身份 API 服务器通信时，我们尝试重新连接的次数。
include_service_catalog = True	布尔值	（可选）指示是否设置 X-Service-Catalog 标头。如果为 False，则中间件不会要求在令牌验证时请求服务目录，且不会设置 X-Service-Catalog 标头。
insecure = False	布尔值	验证 HTTPS 连接。

配置选项 = 默认值	类型	Description
interface = internal	字符串值	用于 Identity API 端点的接口。有效值为 "public", "internal" (默认) 或 "admin"。
keyfile = None	字符串值	如果身份服务器需要客户端证书, 则需要此项
memcache_pool_conn_get_timeout = 10	整数值	(可选) 操作等待从池中获取 memcached 客户端连接的秒数。
memcache_pool_dead_retry = 300	整数值	(可选) 在再次尝试前 memcached 服务器被视为 dead 的秒数。
memcache_pool_maxsize = 10	整数值	(可选) 到每个 memcached 服务器的最大打开连接数。
memcache_pool_socket_timeout = 3	整数值	(可选) 与 memcached 服务器通信的套接字超时 (以秒为单位)。
memcache_pool_unused_timeout = 60	整数值	(可选) 与 memcached 的连接在池关闭前未使用的秒数。
memcache_secret_key = None	字符串值	(可选, 如果定义了 memcache_security_strategy) 此字符串用于密钥派生。
memcache_security_strategy = None	字符串值	(可选) 如果定义, 则指示令牌数据是否应进行身份验证或加密。如果 MAC, 令牌数据在缓存中进行身份验证 (使用 HMAC)。如果 ENCRYPT, 令牌数据会在缓存中加密并进行身份验证。如果该值不是这些选项之一或为空, 则 auth_token 会在初始化时引发异常。
memcache_use_advanced_pool = False	布尔值	(可选) 使用 advanced (事件安全) memcached 客户端池。高级池只能在 python 2.x 下工作。
memcached_servers = None	列表值	(可选) 指定要用于缓存的 memcached 服务器列表。如果未定义, 令牌将会被缓存在进程中。
region_name = None	字符串值	找到身份服务器的区域。
service_token_roles = ['service']	列表值	需要出现在服务令牌中的角色。服务令牌可以请求可以使用过期的令牌, 因此此检查应严格控制只有实际服务应发送此令牌。此处的角色作为 ANY 检查应用, 因此此列表中的任何角色都必须存在。为了向后兼容的原因, 这目前只会影响 allow_expired 检查。
service_token_roles_required = False	布尔值	为了向后兼容的原因, 我们必须允许有效的服务令牌传递, 但不允许将 service_token_roles 检查视为有效。在以后的发行版本中设置此 true 将成为默认设置, 并应启用 (如果可能)。

配置选项 = 默认值	类型	Description
service_type = None	字符串值	服务的名称或类型（在服务目录中出现）。这用于验证具有受限访问规则的令牌。
token_cache_time = 300	整数值	为了防止过度努力验证令牌，中间件会缓存之前可预见的令牌（以秒为单位）。设置为 -1 可完全禁用缓存。
www_authenticate_uri = None	字符串值	完整的"公共"身份 API 端点。此端点不应是一个 "admin" 端点，因为它应该可以被所有最终用户访问。未经身份验证的客户端重定向到此端点进行身份验证。虽然理想情况下，这个端点应该被未指定版本，但通配符中的客户端支持会有所不同。如果您在这里使用版本化的 v2 端点，则该服务用户不应该用于验证令牌的端点，因为普通用户可能无法访问该端点。

9.1.26. libvirt

下表概述了 `/etc/nova/nova.conf` 文件中的 `[libvirt]` 组下可用的选项。

表 9.25. libvirt

配置选项 = 默认值	类型	Description
<code>connection_uri = `</code>	字符串值	<p>覆盖所选虚拟化类型的默认 libvirt URI。</p> <p>如果设置，Nova 将使用此 URI 连接到 libvirt。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 类似于 <code>qemu:///system</code> 的 URI。 <p>This is only necessary if the URI differs to the commonly known URIs for the chosen virtualization type.</p> <p>相关选项：</p> <ul style="list-style-type: none"> <code>virt_type</code>：这里将什么用作默认值。

配置选项 = 默认值	类型	Description
cpu_mode = None	字符串值	<p>用于设置实例应具有 CPU 模式。</p> <p>如果 virt_type="kvm&verbar;qemu"，它将默认为 host-model，否则它将默认为 none。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● cpu_models：这应只在 cpu_mode 设置为 custom 时设置。否则，这会导致错误，实例启动将失败。
cpu_model_extra_flags = []	列表值	<p>启用或禁用客户机 CPU 标记。</p> <p>要显式启用或禁用 CPU 标记，请使用 +flag 或 -flag。-+ 为客户及启用 CPU 标记，而- 会禁用它。如果没有指定 + 或-，则会启用该标志，这是默认的行为。例如，如果您指定了以下内容（假设主机硬件和软件支持上述 CPU 模型和功能）：</p> <pre>[libvirt] cpu_mode = custom cpu_models = Cascadelake-Server cpu_model_extra_flags = -hle, -rtm, +ssbd, mtrr</pre> <p>Nova 将禁用客户机的 hle 和 rtm 标志；并且它将启用 ssbd 和 mttr（因为它使用既没有 + 或- 前缀指定）。</p> <p>CPU 标记不区分大小写。在以下示例中，客户机将禁用 pdpe1gb 标志；vmx 和 pcid 标志将启用：</p> <pre>[libvirt] cpu_mode = custom cpu_models = Haswell-noTSX-IBRS cpu_model_extra_flags = -PDPE1GB, +VMX, pcid</pre> <p>指定额外的 CPU 标志与 cpu_mode config 属性的所有三个可能值有效：custom（这还需要通过 cpu_models 配置属性）、host-models 配置属性指定显式 CPU 模型、host-model 或 host-passthrough。</p> <p>在某些情况下，您可能需要为 host-passthrough CPU 模式配置额外的 CPU 标志，因为有时 QEMU 可能会禁用某些 CPU 功能。例如，Intel 的 "invtsc" (Invariant Time Stamp Counter) CPU 标记是，您需要将此标志公开给 Nova 实例，您需要明确启用它。</p> <p>cpu_model_extra_flags 的可能值取决于使用的</p>

配置选项 = 默认值	类型	CPU 型号。有关给定 CPU 模型的可能 CPU 功能标记，请参阅 <code>/usr/share/libvirt/cpu_mapVRF.xml</code> 。 Description
		<p>特定 CPU 标记的特别备注：<code>pcid</code>（在应用 <i>Meltdown</i> CVE 修复后，降低客户机性能下降）。当使用自定义 CPU 模式配置此标志时，并非所有 CPU 型号（由 QEMU 和 libvirt 定义）都需要它：</p> <ul style="list-style-type: none"> • 包括 <code>pcid</code> 功能的唯一虚拟 CPU 型号是 Intel "Haswell"、"Broadwell" 和 "Skylake" 变体。 • libvirt / QEMU CPU 型号 "Nehalem", "Westmere", "SandyBridge", 和 "IvyBridge" 将不会公开 <code>pcid</code> 功能，即使主机 CPU 具有相同名称包括它。i.e.在使用上述虚拟 CPU 型号时，需要明确指定 <i>PCID</i>。 <p>libvirt 驱动程序的默认 CPU 模式 <code>host-model</code> 将对客户机的 <i>PCID</i> CPU 标志进行正确的操作，假设您正在运行更新的处理器 <code>microcode</code>、主机和客户机内核、libvirt 和 QEMU。其他模式 <code>host-passthrough</code> 检查硬件中是否有 <i>PCID</i>，如果是，则直接将其传递给 Nova 客户机。因此，在 <i>PCID</i> 的上下文中，这些 CPU 模式(<code>host-model</code> 或 <code>host-passthrough</code>)不需要使用 <code>cpu_model_extra_flags</code>。</p> <p>相关选项：</p> <ul style="list-style-type: none"> • <code>cpu_mode</code>

配置选项 = 默认值	类型	Description cpu_models
cpu_models = []	列表值	<p>主机支持的 CPU 模型列表。</p> <p>列表应该被排序，以便在前面列出更常见的和不太高级 CPU 型号。以下是一个示例：SandyBridge,IvyBridge,Haswell,Broadwell，后者的 CPU 模型的功能更丰富的了以前的 CPU 模型。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 命名的 CPU 模型可以通过 virsh cpu-models ARCH 找到，其中 ARCH 是您的主机架构。 <p>相关选项：</p> <ul style="list-style-type: none"> cpu_mode：当您要配置（通过 cpu_models）特定名为 CPU 模型时，应将其设置为 custom。否则，这会导致错误，实例启动将失败。 virt_type：只有虚拟化类型 kvm 和 qemu 使用此方法。 <ul style="list-style-type: none"> a. 注意：请小心，仅指定可以在硬件中完全支持的型号。
device_detach_attempts = 8	整数值	<p>驱动程序尝试在 libvirt 中分离设备的最大尝试次数。</p> <p>相关选项：</p> <ul style="list-style-type: none"> <code>:oslo.config:option:libvirt.device_detach_timeout</code>
device_detach_timeout = 20	整数值	<p>驱动程序等待成功或 libvirt 的失败事件在重新触发分离前尝试的最大秒数。</p> <p>相关选项：</p> <ul style="list-style-type: none"> <code>:oslo.config:option:libvirt.device_detach_attempts</code>
disk_cachemodes = []	列表值	<p>用于不同磁盘类型的特定缓存模式。</p> <p>例如： file=directsync,block=none,network=writeback</p> <p>对于本地或直接附加存储，建议您使用 writethrough（默认）模式，因为它确保数据完整性，并为客户机中运行的应用程序具有可接受的 I/O 性能，特别是读</p>

配置选项 = 默认值	类型	Description
		<p>操作。但是，对于远程 NFS 存储，建议使用缓存模式，因为直接 I/O 操作(O_DIRECT)的性能优于同步 I/O 操作（使用 O_SYNC）。缓存模式不会有效地将所有客户机 I/O 操作转换为主机上的直接 I/O 操作，这是此环境中的 NFS 客户端。</p> <p>可能的缓存模式：</p> <ul style="list-style-type: none"> ● 默认：“取决于具体情况” – 对于 Nova 管理的磁盘，none，如果主机文件系统符合 Linux 的 O_DIRECT 规格；否则为 writeback。对于卷驱动程序，默认决定于具体的驱动程序：除了 SMBFS 和 Virtuzzo（使用 writeback）外，都为 none。 ● none：将缓存模式设置为 none，主机页面缓存被禁用，但为客户机启用了磁盘写入缓存。在这个模式中，客户机中的写入性能是最佳的，因为写入操作会绕过主机页面缓存并直接进入磁盘写入缓存。如果磁盘写缓存被电池支持，或者客户机传输数据中的应用程序或存储堆栈正确传输数据（通过 fsync 操作或文件系统障碍），则可以确保数据完整性。但是，由于主机页面缓存被禁用，因此客户机中的读取性能与启用主机页面缓存的模式（如 writethrough 模式）不佳。可共享磁盘设备，如多重附加块存储卷，无论配置如何，它们的缓存模式都设为 none。 ● writethrough：将缓存设置为 writethrough 模式，则会启用主机页面缓存，但客户机禁用磁盘写入缓存。因此，即使客户机中的应用程序和存储堆栈无法正确将数据传送到持久性存储（通过 fsync 操作或文件系统障碍），此缓存模式也可确保数据完整性。由于在这个模式中启用了主机页面缓存，因此客户机中运行的应用程序的读取性能通常更好。但是，写入性能可能会降低，因为磁盘写入缓存被禁用。 ● writeback：当缓存设置为 writeback 模式时，客户机都启用了主机页面缓存和磁盘写入缓存。因此，客户机中运行的应用程序的 I/O 性能很好，但数据在电源故障时不受保护。因此，建议只针对潜在的数据丢失的临时数据进行这个缓存模式。注意：包含后端磁盘机制可以提供安全回写缓存语义。专门用于绕过主机页面缓存，如 QEMU 的集成式 RBD 驱动程序。Ceph 文档建议将其设置为 writeback 以获得最大性能，同时保持数据安全。 ● directsync：与"writethrough"类似，但它绕过主机页面缓存。 ● insecure：不安全缓存模式可完全忽略缓存传输操作。由于其名称意味着，此缓存模式应该只用于数据丢失的临时数据。这个模式可用于加快客户机安装，但您应该在生产环境中切换到另一个缓存模式。

配置选项 = 默认值	类型	Description
disk_prefix = None	字符串值	<p>覆盖附加到实例的设备的默认磁盘前缀。</p> <p>如果设置，这将用来识别总线的可用磁盘空间名称。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何生成有效磁盘设备名称（如 <i>sda</i> 或 <i>hda</i>）的前缀，例如：只有在设备名称与虚拟化类型的常用设备名称前缀不同（如 <i>sd</i>、<i>xvd</i>、<i>uud</i>、<i>vd</i>）时才需要。 <p>相关选项：</p> <ul style="list-style-type: none"> virt_type：使用哪个设备类型，这决定了默认的磁盘前缀。
enabled_perf_events = []	列表值	<p>监视和收集统计数据性能事件。</p> <p>这将允许您指定事件列表来监控客户机的低级别性能，并通过 <i>libvirt</i> 驱动程序收集相关统计信息，后者又使用 Linux 内核的 perf 基础架构。设置此配置属性后，Nova 将生成 <i>libvirt</i> 客户机 XML 来监控指定的事件。</p> <p>例如，要监控 CPU 周期(<i>total/elapsed</i>)和缓存未命中的计数，请按如下所示启用它们：</p> <pre>[libvirt] enabled_perf_events = cpu_clock, cache_misses</pre> <p>可能的值：字符串列表。支持的事件列表包括在这里__。请注意，Intel CMT 事件 - 'cmt、mbmbt 和 mbml - 由最新的 Linux 内核版本(4.14+)不支持，nova 会忽略。</p> <p>— https://libvirt.org/formatdomain.html#elementsPerf.</p>

配置选项 = 默认值	类型	Description
file_backed_memory = 0	整数值	<p>可用容量（以 MiB 为单位）用于文件支持的内存。</p> <p>设置为 0 可禁用文件支持的内存。</p> <p>启用后，实例将在 /etc/libvirt/qemu.conf 的 memory_backing_dir 选项指定的目录中创建内存文件。默认位置为 /var/lib/libvirt/qemu/ram。</p> <p>启用后，为此选项定义的值会报告为节点内存容量。计算节点系统内存将通过内核的 pagecache 机制用作文件支持的内存的缓存。</p> <ul style="list-style-type: none"> a. 注意：这个功能与巨页不兼容。 b. 注意：这个功能与内存过量使用不兼容。 <p>相关选项：</p> <ul style="list-style-type: none"> ● virt_type 必须设置为 kvm 或 qemu。 ● ram_allocation_ratio 必须设置为 1.0。
gid_maps = []	列表值	<p>guid 目标和范围列表。Syntax 是 guest-gid:host-gid:count。允许最多 5 个。</p>
hw_disk_discard = None	字符串值	<p>nova 受管磁盘的丢弃选项。</p> <p>需要：</p> <ul style="list-style-type: none"> ● Libvirt >= 1.0.6 ● qemu >= 1.5（原始格式） ● qemu >= 1.6 (qcow2 格式)
hw_machine_type = None	列表值	<p>对于 qemu 或 KVM 客户机，将此选项设置为为每个主机架构指定默认机器类型。您可以通过检查 <code>:command:virsh capabilities</code> 命令的输出来查找环境中支持的机器类型列表。此配置选项的值格式为 host-arch=machine-type。例如：x86_64=machinetype1,armv7l=machinetype2。</p>
<code>`images_rbd_ceph_conf = `</code>	字符串值	<p>要使用的 ceph 配置文件的路径</p>

配置选项 = 默认值	类型	Description
images_rbd_glance_copy_poll_interval = 15	整数值	<p>要求将镜像复制到本地 rbd 存储后轮询 Glance 的时间间隔（以秒为单位）。这会影响我们要求 Glance 报告复制完成的频率，因此应该很快注意到这个问题，但我们不会给 Glance 服务器产生意外负载。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● images_type - 必须设置为 rbd ● images_rbd_glance_store_name - 必须设置为存储名称
images_rbd_glance_copy_timeout = 600	整数值	<p>我们将等待 Glance 完成镜像副本到本地 rbd 存储的总最长时间。这应该足够长，允许将大型镜像复制到本地存储之间的网络链接和镜像通常所在的位置。设置的不足时间只是捕获镜像副本停滞或继续太慢时很有用的情况。Glance 将报告实际错误，并根据轮询间隔进行通知。</p> <p>相关选项：* images_type - 必须设置为 rbd * images_rbd_glance_store_name - 必须设置为存储名称 * images_rbd_glance_copy_poll_interval - 控制故障 time-to-notice</p>
<code>images_rbd_glance_store_name = `</code>	字符串值	<p>代表此节点使用的 rbd 集群的 Glance 存储的名称。如果设置，这将允许 Nova 请求 Glance 将现有的非本地存储中的镜像复制到此选项命名的位置，以便维护正确的写时复制行为。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● images_type - 必须设置为 rbd ● images_rbd_glance_copy_poll_interval - 控制状态轮询频率 ● images_rbd_glance_copy_timeout - 控制整个复制超时
images_rbd_pool = rbd	字符串值	存储 rbd 卷的 RADOS 池

配置选项 = 默认值	类型	Description
images_type = default	字符串值	<p>VM 镜像格式。</p> <p>如果指定了 default，则使用 use_cow_images 标志而不是这个标志。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● compute.use_cow_images ● images_volume_group ● [workarounds]/ensure_libvirt_rbd_instance_dir_cleanup ● compute.force_raw_images
images_volume_group = None	字符串值	<p>当您指定 images_type=lvm 时，用于虚拟机镜像的 LVM 卷组</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● images_type
inject_key = False	布尔值	<p>允许在引导时注入 SSH 密钥。</p> <p>镜像中不需要代理来执行此操作。如果主机上的 libguestfs 可用，它将被使用。否则会使用 nbd。镜像的文件系统将被挂载，SSH 密钥（在 REST API 调用中提供的）将作为 root 用户的 SSH 密钥注入，并附加到该用户的 authorized_keys。如有必要，将设置 SELinux 上下文。请注意，当实例从卷启动时，无法 注入。</p> <p>此配置选项将启用直接修改实例磁盘，不会影响 cloud-init 可使用 config_drive 选项或元数据服务的数据。</p> <p>仅限 Linux 发行版客户机。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● inject_partition：此选项将决定文件系统的发现和使用。它还可以禁用注入。

配置选项 = 默认值	类型	Description
inject_partition = -2	整数	<p>决定如何选择文件系统来注入数据。</p> <p>libguestfs 用于注入数据。如果 libguestfs 无法确定 root 分区（因为有多于一个或小于一个根分区），或者无法挂载文件系统，则会导致错误，实例无法引导。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● -2 especially 禁用数据的注入。 ● -1 jpeg 查找使用 libguestfs 挂载的文件系统的根分区 ● 0 omitted the image is not partitioned ● >0 TOKEN 用于注入的分区数量 <p>仅限 Linux 发行版客户机。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● inject_key：如果此选项允许注入它依赖于值大于或等于 -1（用于 inject_partition）的 SSH 密钥。 ● inject_password：如果此选项允许注入 admin 密码，它依赖于值大于或等于 -1（用于 inject_partition）。 ● [guestfs]/debug 您可以使用这个配置选项启用 debug 日志级别 libguestfs。调试问题时将有更详细的输出帮助。 ● virt_type：如果您使用 lxc 作为 virt_type，它将被视为单个分区镜像
inject_password = False	布尔值	<p>仅允许在创建和重新构建过程中为实例注入管理员密码。</p> <p>镜像中不需要代理来执行此操作。如果主机上的 libguestfs 可用，它将被使用。否则会使用 nbd。镜像的文件系统将被挂载，以及 REST API 调用中提供的 admin 密码，该密码将作为 root 用户的密码注入。如果没有 root 用户可用，实例将不会启动，并且抛出错误。请注意，当实例从卷启动时，无法注入。</p>

配置选项 = 默认值	类型	Description 以及 Linux 发行版客户机。
		<p>可能的值：</p> <ul style="list-style-type: none"> • true：允许注入。 • false：不允许注入。任何通过 REST API 提供的 admin 密码都会静默忽略。 <p>相关选项：</p> <ul style="list-style-type: none"> • inject_partition：此选项将决定文件系统的发现和使用。它还可以禁用注入。
iscsi_iface = None	字符串值	<p>如果需要卸载支持，用于连接到目标的 iSCSI 传输 iface。</p> <p>默认格式为 <code><transport_name>.<hwaddress></code>，其中 <code><transport_name></code> 是接口之一 (<code>be2iscsi,bnx2i,cxgb3i,cxgb4i,qla4xxx,ocs,tcp</code>) 和 <code><hwaddress></code> 是接口的 MAC 地址，可以通过 <code>iscsiadm -m iface</code> 命令生成。不要将 <code>iscsi_iface</code> 参数与实际传输名称混淆。</p>
iser_use_multipath = False	布尔值	<p>使用 iSER 卷的多路径连接。</p> <p>iser 卷可以连接为多路径设备。这将提供高可用性和容错功能。</p>

配置选项 = 默认值	类型	Description
live_migration_bandwidth = 0	整数值	<p>迁移期间使用的最大带宽（单位 MiB/s）。</p> <p>如果设置为 0，则虚拟机监控程序将选择合适的默认值。有些虚拟机监控程序不支持此功能，如果带宽不是 0，则返回错误。详情请查看 libvirt 文档。</p>
live_migration_completion_timeout = 800	整数值	<p>迁移在中止操作前成功完成传输数据的时间（以秒为单位）。</p> <p>值是每个要传输的客户机 RAM + 磁盘，其绑定至少为 2 GiB。通常应该大于停机延迟 * 停机时间步骤。设置为 0 以禁用超时。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● live_migration_downtime ● live_migration_downtime_steps ● live_migration_downtime_delay
live_migration_downtime = 500	整数值	<p>实时迁移切换的最大允许停机时间（以毫秒为单位）。</p> <p>将向上舍入到至少 100ms。如果要允许实时迁移更快地完成，或者通过允许客户机暂停以更长时间进行实时迁移切换期间，您可以增加这个值。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● live_migration_completion_timeout
live_migration_downtime_delay = 75	整数值	<p>迁移停机时间增加每个步骤之间等待的时间（以秒为单位）。</p> <p>最小延迟为 3 秒。值是每个要传输的客户机 RAM + 磁盘，其绑定至少为 2 GiB 的设备。</p>
live_migration_downtime_steps = 10	整数值	<p>达到最大停机时间值的增量步骤数。</p> <p>将向上舍入到至少 3 个步骤。</p>

配置选项 = 默认值	类型	Description
live_migration_inbound_address = None	主机地址值	<p>用作此主机的实时迁移地址的 IP 地址。</p> <p>此选项表示在迁移到此 hypervisor 时应用作实时迁移流量的 IP 地址。然后，实时迁移流量源使用此元数据来构建迁移 URI。</p> <p>如果此选项设置为 None，则使用迁移目标计算节点的主机名。</p> <p>这个选项对于在实时迁移流量可能会显著影响 network plane 的环境中非常有用。然后，用于实时迁移流量的单独网络可以使用这个配置选项并避免对管理网络的影响。</p>
live_migration_permit_auto_converge = False	布尔值	<p>此选项允许 nova 使用自动聚合来启动实时迁移。</p> <p>如果实时迁移速度缓慢，自动聚合会减慢 CPU。只有在此标志被设置为 True 且因为使用的 libvirt 和 QEMU 的版本不可用时，才会使用自动聚合。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● live_migration_permit_post_copy
live_migration_permit_post_copy = False	布尔值	<p>这个选项允许 nova 将当前实时迁移切换到 post-copy 模式，即，在迁移完成前，将活动虚拟机切换到目标节点上的一个大小，从而确保需要传输的内存上的上限。post-copy 需要 libvirt>=1.3.3 和 QEMU>=2.5.0。</p> <p>允许后，如果达到 live_migration_completion_timeout 定义的超时，且，则 live_migration_timeout_action 设置为 <i>force_complete</i>，则 post-copy 模式将自动激活。请注意，如果您更改为没有超时，或选择使用 <i>abort</i>，即 live_migration_completion_timeout = 0，则不会自动切换到 post-copy。</p> <p>实时迁移强制完成 API 在允许后使用 post-copy。如果 post-copy 模式不可用，则强制完全回退到暂停虚拟机，以确保实时迁移操作完成。</p> <p>使用 post-copy 模式时，如果源和目标主机丢失网络连接，则需要重启实时迁移的虚拟机。如需了解更多信息，请参阅管理指南。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● live_migration_permit_auto_converge ● live_migration_timeout_action

配置选项 = 默认值	类型	Description
live_migration_scheme = None	字符串值	<p>实时迁移流量来源使用的实时迁移的 URI 方案。</p> <p>覆盖默认的 libvirt 实时迁移方案（依赖于 virt_type）。如果此选项设置为 None，nova 将根据虚拟机监控程序自动选择合理默认值。不建议您更改此设置，除非您确定 hypervisor 支持特定方案。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● virt_type：只有在 virt_type 设为 kvm 或 qemu 时，此选项才有意义。 ● live_migration_uri: 如果 live_migration_uri 值不是 None，则用于实时迁移的方案将从 live_migration_uri 替代。
live_migration_timeout_action = abort	字符串值	<p>这个选项将用于决定在 live_migration_completion_timeout 过期后对虚拟机采取什么操作。默认情况下，实时迁移操作将在完成超时后中止。如果设置为 force_complete，则计算服务将根据启用了 post copy 且可用 (live_migration_permit_post_copy 设置为 True) 暂停虚拟机或触发后复制。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● live_migration_completion_timeout ● live_migration_permit_post_copy
live_migration_tunnelled = False	布尔值	<p>启用隧道迁移。</p> <p>这个选项启用隧道迁移功能，其中迁移数据通过 libvirtd 连接传输。如果启用，我们使用 VIR_MIGRATE_TUNNELLED 迁移标志，避免了需要配置网络，以允许直接 hypervisor 到 hypervisor 通信。如果为 False，请使用原生传输。如果没有设置，Nova 将根据在虚拟机监控程序中选择本地加密支持的可用性。启用此选项会绝对影响性能。</p> <p>请注意，这个选项与块迁移的使用不兼容。</p> <p>弃用自：23.0.0</p> <p>原因： "tunnelled 实时迁移" 有两个固有的限制：它无法处理非共享存储设置中的磁盘的实时迁移；它具有巨大的性能成本。这两个问题都由 "live_migration_with_native_tls"（需要预先配置的 TLS 环境）解决，这是保护所有实时迁移流的建议方法。</p>

配置选项 = 默认值	类型	Description
live_migration_uri = None	字符串值	<p>实时迁移流量源使用的实时迁移目标 URI。</p> <p>覆盖默认的 libvirt 实时迁移目标 URI（依赖于 virt_type）。任何包含的"%s"都会被迁移目标主机名替换，如果设为 live_migration_inbound_addr，则替换 live_migration_inbound_addr。</p> <p>如果此选项被设置为 None（默认值），Nova 将自动根据以下列表中支持的 4 个 virt_type 生成 live_migration_uri 值：</p> <ul style="list-style-type: none"> ● <i>kvm: qemu+tcp://%s/system</i> ● <i>qemu: qemu+tcp://%s/system</i> ● <i>parallels: parallels+tcp://%s/system</i> <p>相关选项：</p> <ul style="list-style-type: none"> ● live_migration_inbound_addr: 如果 live_migration_inbound_addr 值不是 None，并且 live_migration_tunneled 为 False，将使用目标计算节点的 ip/hostname 地址而不是 live_migration_uri 作为作实时迁移的 uri。 ● live_migration_scheme: 如果没有设置 live_migration_uri，则用于实时迁移的方案将从 live_migration_scheme 替代。 <p>弃用自：15.0.0</p> <p>reason: live_migration_uri 已弃用，它替代了允许更改实时迁移方案和目标 URI: "live_migration_scheme" 和 "live_migration_inbound_addr" 的其它选项。</p>

配置选项 = 默认值	类型	Description
live_migration_with_native_tls = False	布尔值	<p>在实时迁移时使用 QEMU 原生 TLS 加密。</p> <p>这个选项将允许迁移流（客户机 RAM 和设备状态）和磁盘流通过原生 TLS 传输，例如，内置于 QEMU 中的 TLS 支持。</p> <p>先决条件：在所有相关的 Compute 节点上正确配置 TLS 环境。这意味着，认证机构(CA)、服务器、客户端证书、对应的密钥及其文件已发生，并验证。</p> <p>备注：</p> <ul style="list-style-type: none"> ● 为了对迁移流和磁盘流（也称为 "block migration"）进行加密，live_migration_with_native_tls 是首选的配置属性，而不是 live_migration_tunnelled。 ● 由于两个主要原因，live_migration_tunnelled 将在长期弃用：(a)会导致大量性能损失；(b)它与块迁移不兼容。因此，如果您的计算节点至少具有 libvirt 4.4.0 和 QEMU 2.11.0，则强烈建议您使用 live_migration_with_native_tls。 ● live_migration_tunnelled 和 live_migration_with_native_tls 不应同时使用。 ● 与 live_migration_tunnelled 不同，live_migration_with_native_tls 与块迁移兼容。也就是说，使用这个选项时，NBD 流将加密哪些磁盘迁移到目标主机。 <p>相关选项：</p> <p>live_migration_tunnelled：通过 libvirtd 传输迁移流（而不是磁盘流）。</p>
max_queues = None	整数值	<p>创建多队列客户端时可启用的最大 virtio 队列对数。分配的 virtio 队列数量将减少客户机所请求的 CPU 以及定义的最大值。默认情况下，这个值被设置为 none，这意味着将使用根据所报告的内核主版本的传统限制。</p>
mem_stats_period_seconds = 10	整数值	<p>内存用量统计周期的秒数。零或负值意味着禁用内存用量统计。</p>

配置选项 = 默认值	类型	Description
nfs_mount_options = None	字符串值	<p>传递给 NFS 客户端的挂载选项。详情请查看 <code>nfs man page</code> 中的部分。</p> <p>挂载选项控制挂载文件系统的方式，以及 NFS 客户端在访问此挂载点上的文件时的行为。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表以逗号分开的挂载选项的任何字符串。 ● 字符串示例：<code>vers=3,lookupcache=pos</code>
nfs_mount_point_base = \$state_path/mnt	字符串值	<p>挂载到计算节点上 NFS 卷的目录。默认为安装 nova 的 Python 模块的 <code>mnt</code> 目录。</p> <p>NFS 为 OpenStack 块存储服务提供共享存储。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表挂载点的绝对路径的字符串。
num_aoe_discover_tries = 3	整数值	<p>重新发现 AoE 目标以查找卷的次数。</p> <p>Nova 支持通过 AOE (ATA 通过以太网)连接到主机的块存储。这个选项允许用户指定可发现 AoE 设备的最大重试尝试数。</p>
num_iser_scan_tries = 5	整数值	<p>扫描 iSER 目标以查找卷的次数。</p> <p>iser 是一个服务器网络协议，它扩展了 iSCSI 协议以使用远程直接内存访问(RDMA)。此选项允许用户指定可用于查找 iSER 卷的最大扫描尝试次数。</p>

配置选项 = 默认值	类型	Description
num_memory_encrypted_guests = None	整数值	<p>具有加密内存的最大客户机数，可在此计算主机上同时运行。</p> <p>现在，这只适用于支持 SEV（安全加密虚拟化）的 AMD 机器。这些机器在其内存控制器中有有限数量的插槽来存储加密密钥。每个运行带有加密内存的客户机都将消耗其中一个插槽。</p> <p>未来可能会将选项用于其他同等技术。如果机器不支持内存加密，则忽略该选项并将清单设置为 0。</p> <p>如果机器确实支持内存加密，对于 None 值，则代表没有限制，例如，Nova 不会对可启动的 SEV 客户机数量施加任何限制，即使底层硬件会强制实施自己的限制。但是，在未来预期将来，从硬件的自动检测清单将变得可能，此时 None 将导致自动检测自动进行正确的限制。</p> <p>a. note::</p> <div style="border-left: 2px solid black; padding-left: 10px; margin-left: 20px;"> <p>It is recommended to read :ref:`the deployment documentation's section on this option <num_memory_encrypted_guests>` before deciding whether to configure this setting or leave it at the default.</p> </div> <p>相关选项：</p> <ul style="list-style-type: none"> • :oslo.config:option:libvirt.virt_type 必须设置为 kvm。 • 建议考虑在 :oslo.config:option:libvirt.hw_machine_type; :ref: deploy-sev-enabled-infrastructure 中包括 x86_64=q35。
num_nvme_discover_tries = 5	整数值	<p>重新发现 NVMe 目标来查找卷的次数</p> <p>Nova 支持通过 NVMe 附加到主机的块存储(Non-Volatile Memory Express)。这个选项允许用户指定可发现 NVMe 设备的最大重试尝试数。</p>

配置选项 = 默认值	类型	Description
num_pcie_ports = 0	整数值	<p>实例将获取的 PCIe 端口数量。</p> <p>libvirt 允许一个目标实例的自定义 PCIe 端口 (pcie-root-port 控制器)。其中一些将默认使用, 其余则可用于热插拔。</p> <p>默认情况下, 我们只有 1-2 个可用端口来限制热插拔。</p> <p>更多信息 : https://github.com/qemu/qemu/blob/master/docs/pcie.txt</p> <p>由于 aarch64/virt 最大值的 QEMU 限制设置为 28。</p> <p>默认值 0 将端口数量移到 libvirt。</p>
num_volume_scan_tries = 5	整数值	<p>扫描给定存储协议以查找卷的次数。</p>
pmem_namespaces = []	列表值	<p>配置持久内存(pmem)命名空间。</p> <p>这些命名空间必须已在主机上创建。这个配置选项采用以下格式 :</p> <pre>"\$LABEL:\$NSNAME[&verbar;\$NSNAME] [, \$LABEL:\$NSNAME[&verbar;\$NSNAME]]"</pre> <ul style="list-style-type: none"> ● \$NSNAME 是 pmem 命名空间的名称。 ● \$LABEL 代表一个资源类, 这用于生成资源类名称为 CUSTOM_PMEM_NAMESPACE_\$LABEL。 <p>例如 :</p> <pre>[libvirt] pmem_namespaces=128G:ns0 ns1 ns2 ns3,262144MB:ns4 ns5,MEDIUM:ns6 ns7</pre>
quobyte_client_cfg = None	字符串值	<p>指向 Quobyte 客户端配置文件的路径。</p>
quobyte_mount_point_base = \$state_path/mnt	字符串值	<p>在计算节点上挂载 Quobyte 卷的目录。</p> <p>Nova 支持 Quobyte 卷驱动程序, 允许在 Quobyte 存储后端上存储块存储服务卷。此选项指定挂载 Quobyte 卷的目录路径。</p> <p>可能的值 :</p> <ul style="list-style-type: none"> ● 代表挂载点的绝对路径的字符串。

配置选项 = 默认值	类型	Description
rdp_connect_timeout = 5	整数值	最初连接到集群时 RADOS 客户端超时（以秒为单位）。
rdp_destroy_volume_retries = 12	整数值	销毁 RBD 卷的重试次数。 相关选项： <ul style="list-style-type: none"> • [libvirt]/images_type = <i>rdp</i>
rdp_destroy_volume_retry_interval = 5	整数值	连续重试间等待的秒数，以销毁 RBD 卷。 相关选项： <ul style="list-style-type: none"> • [libvirt]/images_type = <i>rdp</i>
rdp_secret_uuid = None	字符串值	rdp_user 卷的 secret 的 libvirt UUID。
rdp_user = None	字符串值	用于访问 rbd (RADOS 块设备)卷的 RADOS 客户端名称。 通过 Ceph RBD 服务器连接和验证时，libvirt 将引用此用户。
realtime_scheduler_priority = 1	整数值	在客户机的实时主机上下文 vCPU 中，将在该调度优先级中运行。优先级取决于主机内核（通常为 1-99）
remote_filesystem_transport = ssh	字符串值	libvirt 远程文件操作的传输方法。 由于 libvirt 无法使用 RPC 通过网络复制文件到/从其他计算节点复制文件，所以必须使用其他方法： <ul style="list-style-type: none"> • 在远程主机上创建目录 • 在远程主机上创建文件 • 从远程主机中删除文件 • 将文件复制到远程主机

配置选项 = 默认值	类型	Description
rescue_image_id = None	字符串值	<p>从中引导到来自损坏实例的救援数据的镜像的 ID。</p> <p>如果 rescue REST API 操作不提供要使用的镜像的 ID，则会使用此 ID 引用的镜像。如果没有设置这个选项，则使用来自实例的镜像。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 镜像或任何内容的 ID。如果它指向 Amazon Machine Image (AMI)，请考虑设置配置选项 rescue_kernel_id 和 rescue_ramdisk_id。如果没有设置，则使用实例的镜像。 <p>相关选项：</p> <ul style="list-style-type: none"> ● rescue_kernel_id：如果所选的救援镜像允许单独定义其内核磁盘，则使用这个选项的值（如果指定）。当 Amazon 的 AMI/AKI/ARI 镜像格式用于救援镜像时，会出现这种情况。 ● rescue_ramdisk_id：如果所选的救援镜像允许分隔的 RAM 磁盘定义，则使用此选项的值（如果指定）。当 Amazon 的 AMI/AKI/ARI 镜像格式用于救援镜像时，会出现这种情况。
rescue_kernel_id = None	字符串值	<p>与救援镜像一起使用的内核(AKI)镜像的 ID。</p> <p>如果所选的救援镜像允许单独定义其内核磁盘，则使用这个选项的值（如果指定）。当 Amazon 的 AMI/AKI/ARI 镜像格式用于救援镜像时，会出现这种情况。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 内核镜像的 ID 或任何内容。如果未指定，则使用实例中的内核磁盘（如果实例启动）。 <p>相关选项：</p> <ul style="list-style-type: none"> ● rescue_image_id：如果该选项指向 Amazon 的 AMI/AKI/ARI 镜像格式中的镜像，则也很有用使用 rescue_kernel_id。

配置选项 = 默认值	类型	Description
rescue_ramdisk_id = None	字符串值	<p>与救援镜像一起使用的 RAM 磁盘(ARI)镜像的 ID。</p> <p>如果所选的救援镜像允许单独定义其 RAM 磁盘，则使用这个选项的值（如果指定）。当 Amazon 的 AMI/AKI/ARI 镜像格式用于救援镜像时，会出现这种情况。</p> <p>可能的值：</p> <ul style="list-style-type: none"> RAM 磁盘镜像或无任何内容的 ID。如果未指定，则使用实例的 RAM 磁盘（如果实例启动）。 <p>相关选项：</p> <ul style="list-style-type: none"> rescue_image_id：如果该选项指向 Amazon 的 AMI/AKI/ARI 镜像格式中的镜像，则也很有用处使用 rescue_ramdisk_id。
rng_dev_path = /dev/urandom	字符串值	<p>将用作主机上熵的 RNG（随机数字生成器）设备的路径。从 libvirt 1.3.4 开始，接受任何路径（当读取时返回随机数字）。推荐的熵源是 /dev/urandom criu-wagonit is non-blocking，因此比较快，并避免 /dev/random 的限制，这是旧接口。如需了解更多信息（以及不同 RNG 源之间的比较），请参阅 Linux 内核 API 文档中的"用法"部分</p> <p>[u]random:http://man7.org/linux/man-pages/man4/urandom.4.html 和 http://man7.org/linux/man-pages/man7/random.7.html。</p>
rx_queue_size = None	整数值	<p>配置 virtio rx 队列大小。</p> <p>这个选项仅适用于带有 vhost 和 vhost-user 后端的 virtio-net 设备。仅适用于 QEMU/KVM。需要 libvirt v2.3 QEMU v2.7。</p>
^smbfs_mount_options = ^	字符串值	<p>传递给 SMBFS 客户端的挂载选项。</p> <p>将 SMBFS 选项作为包含所有参数的单一字符串提供。详情请查看 <code>mount.cifs</code> 手册页。请注意，必须指定 libvirt-qemu uid 和 gid。</p>
smbfs_mount_point_base = \$state_path/mnt	字符串值	<p>SMBFS 共享挂载在计算节点上的目录。</p>

配置选项 = 默认值	类型	Description
snapshot_compression = False	布尔值	<p>为 qcow2 镜像启用快照压缩。</p> <p>注：您可以将 snapshot_image_format 设置为 qcow2，以强制所有快照都与原始镜像类型独立使用 qcow2 格式。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● snapshot_image_format
snapshot_image_format = None	字符串值	<p>在发送到镜像服务时确定快照镜像格式。</p> <p>如果设置，这将决定将快照发送到镜像服务时使用的格式。如果没有设置，则默认为与源镜像相同的类型。</p>
snapshots_directory = \$instances_path/snapshots	字符串值	<p>libvirt 驱动程序将存储快照的位置，然后将其上传到镜像服务</p>
sparse_logical_volumes = False	布尔值	<p>如果此标志设为 True，则创建稀疏逻辑卷（使用 virtalsize）。</p> <p>弃用自：18.0.0</p> <p>reason: 稀疏逻辑卷是一个没有测试的功能，因此不被支持。默认情况下，LVM 逻辑卷是预分配的。如果您需要精简配置，请使用 Cinder 精简配置的卷。</p>
swtpm_enabled = False	布尔值	<p>在客户机中启用模拟 TPM（受信任的平台模块）。</p>
swtpm_group = tss	字符串值	<p>swtpm 二进制文件作为运行的组。</p> <p>使用模拟 TPM 时，swtpm 二进制文件将运行来模拟 TPM 设备。此二进制文件以身份运行的用户取决于 libvirt 配置，tss 是默认值。</p> <p>为了支持冷迁移和调整大小，nova 需要知道运行 swtpm 二进制文件的组，以确保文件在节点间移动后获得正确的所有权。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● 还必须设置 swtpm_user。

配置选项 = 默认值	类型	Description
swtpm_user = tss	字符串值	<p>swtpm 二进制文件以。</p> <p>使用模拟 TPM 时，swtpm 二进制文件将运行来模拟 TPM 设备。此二进制文件以身份运行的用户取决于 libvirt 配置，tss 是默认值。</p> <p>为了支持冷迁移和调整大小，nova 需要知道运行 swtpm 二进制文件的用户，以确保文件在节点间移动后获得正确的所有权。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● 还必须设置 swtpm_group。
sysinfo_serial = unique	字符串值	<p>用于填充虚拟 BIOS 中 guest 的主机"serial" UUID 的数据源。除将实例迁移到另一台主机时，除唯一性外，所有选择都会更改串行。更改此选项后，在停止并再次启动后，更改此选项也会影响此主机上的现有实例。建议使用默认选择(唯一)，因为这会在迁移实例时不会改变。但是，如果除了每个实例序列号外，还需要每个主机的序列号，那么请考虑通过主机聚合限制类别。</p>
tx_queue_size = None	整数值	<p>配置 virtio tx 队列大小。</p> <p>这个选项仅适用于带有 vhost-user 后端的 virtio-net 设备。仅适用于 QEMU/KVM。需要 libvirt v3.7 QEMU v2.10。</p>
uid_maps = []	列表值	<p>uid 目标和范围列表。Syntax 是 guest-uid:host-uid:count。允许最多 5 个。</p>
use_virtio_for_bridges = True	布尔值	<p>使用 virtio 作为 KVM/QEMU 的桥接接口</p>

配置选项 = 默认值	类型	Description
virt_type = kvm	字符串值	<p>描述虚拟化类型（或称为域类型）libvirt 应该使用。</p> <p>这个类型的选择必须与您为此主机选择的底层虚拟化策略匹配。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● connection_uri: 取决于这个问题 ● disk_prefix: 取决于这个问题 ● cpu_mode : 这取决于 ● cpu_models: 取决于这个问题
volume_clear = zero	字符串值	<p>在删除临时磁盘时用于擦除临时磁盘的方法。仅在将 LVM 设置为后备存储时才生效。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● images_type - must be set to lvm ● volume_clear_size
volume_clear_size = 0	整数值	<p>MiB 中所分配的卷开始的区域大小，从分配的卷开始，将使用 volume_clear 选项中设置的方法清除。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 0 - 清除整个卷 ● >0 - 清除指定的 MiB <p>相关选项：</p> <ul style="list-style-type: none"> ● images_type - must be set to lvm ● volume_clear - 必须设置，值不是 none 时才能使此选项产生实际影响
volume_use_multipath = False	布尔值	<p>使用 iSCSI 或 FC 卷的多路径连接</p> <p>卷可以在 LibVirt 作为多路径设备中连接。这将提供高可用性和容错功能。</p>

配置选项 = 默认值	类型	Description
vzstorage_cache_path = None	字符串值	<p>SSD 缓存文件的路径。</p> <p>您可以将 SSD 驱动器附加到客户端，并将驱动器配置为存储频繁访问数据的本地缓存。通过在客户端的 SSD 驱动器上有一个本地缓存，您可以将集群的整体性能增加到 10 或更多次。警告！有许多 SSD 模型不是 server grade，并可能对电源丢失进行任意的一组数据更改。此类 SSD 不应在 Vstorage 中使用，并且存在危险，因为可能会导致数据崩溃和不一致。请参考手册，了解哪些 SSD 模型已知是安全的，或使用 <code>vzstorage-hwflush-check (1)</code> 实用程序进行验证。</p> <p>此选项定义应包含 "%(cluster_name)s" 模板的路径，来将缓存与多个共享分开。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>vzstorage_mount_opts</code> 可能包括更详细的缓存选项。
vzstorage_log_path = /var/log/vzstorage/%(cluster_name)s/nova.log.gz	字符串值	<p>vzstorage 客户端日志的路径。</p> <p>此选项定义集群操作的日志，它应包含 "%(cluster_name)s" 模板，以将日志与多个共享分开。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>vzstorage_mount_opts</code> 可能包括更详细的日志记录选项。
vzstorage_mount_group = qemu	字符串值	<p>挂载所有者组名称。</p> <p>此选项定义 Vzstorage 集群挂载点的所有者组。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>vzstorage_mountedServiceSet</code> 组的参数
vzstorage_mount_opts = []	列表值	<p>pstorage-mount 的额外挂载选项</p> <p>有关它们的完整描述，请参阅 https://static.openvz.org/vz-man/man1/pstorage-mount.1.gz.html 格式是参数列表的 python 字符串表示，如 <code>"[-v,-R,500]"</code> Shouldn't include <code>-c, -l, -C, -u, -g</code> 和 <code>-m</code> 作为那些具有显式 <code>vzstorageer</code> 选项的选项。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● 所有其他 <code>vzstorage AlertManager</code> 选项

配置选项 = 默认值	类型	Description
vzstorage_mount_perms = 0770	字符串值	<p>挂载访问模式。</p> <p>此选项定义 Vzstorage 集群挂载点的访问位，其格式类似于 <code>chmod (1)</code>实用程序，如下所示：0770。它由一到四个数字组成，范围从 0 到 7，如果缺少前面的位则代表这些位为 0。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>vzstorage_mountedServiceSet</code> 组的参数
vzstorage_mount_point_base = \$state_path/mnt	字符串值	<p>在计算节点上挂载 Virtuozzo 存储集群的目录。</p> <p>这个选项为 Vzstorage 集群定义非标准挂载点。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>vzstorage_mountedServiceSet</code> 组的参数
vzstorage_mount_user = stack	字符串值	<p>挂载所有者用户名。</p> <p>此选项定义 Vzstorage 集群挂载点的所有者用户。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>vzstorage_mountedServiceSet</code> 组的参数
wait_soft_reboot_seconds = 120	整数值	<p>在发出软重启请求后等待实例关闭的秒数。如果实例在此窗口中没有关闭，我们回退到硬重启。</p>

9.1.27. metrics

下表概述了 `/etc/nova/nova.conf` 文件中的 `[metrics]` 组下可用的选项。

表 9.26. metrics

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
必需 = True	布尔值	<p>是否需要指标。</p> <p>此设置决定了如何处理不可用的指标。如果此选项设置为 True，则指标不可用的任何主机都会引发异常，因此建议在 weighing 前使用 MetricFilter 来过滤这些主机。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 布尔值，其中 False 确保主机的任何指标都不可用，则将主机权重设置为 [metrics] weight_of_unavailable。 <p>相关选项：</p> <ul style="list-style-type: none"> [metrics] weight_of_unavailable
weight_multiplier = 1.0	浮点值	<p>根据报告的指标，用于权衡主机的倍数。</p> <p>当使用指标来权重主机的适用性时，您可以使用此选项更改计算的权重对分配给主机的权重的影响，如下所示：</p> <ul style="list-style-type: none"> >1.0: 增加指标对整体权重的影响 1.0 : 对计算的权重没有改变 >0.0,<1.0: 减少指标对整体权重的影响 0.0 : 指标值会被忽略，并返回 [metrics] weight_of_unavailable 选项的值 >-1.0,<0.0: effect 被减少并撤销 -1.0: 效果被撤销 <-1.0: 效果会按比例和反向增加 <p>可能的值：</p> <ul style="list-style-type: none"> 整数或浮点值，其中值对应于此 weigher 的多条比率。 <p>相关选项：</p> <ul style="list-style-type: none"> [filter_scheduler] weight_classes [metrics] weight_of_unavailable

配置选项 = 默认值	类型	Description
weight_of_unavailable = -10000.0	浮点值	<p>不可用指标的默认权重。</p> <p>当满足以下任一条件时，将使用这个值来代替任何实际指标值：</p> <ul style="list-style-type: none"> ● 主机不能使用 [metrics] weight_setting 命名的一个指标，required 的值为 False。 ● 在 [metrics] weight_setting 中为指标指定的比率为 0。 ● [metrics] weight_multiplier 选项设为 0。 <p>可能的值：</p> <ul style="list-style-type: none"> ● 整数或浮点值，其中值对应于此 weigher 的多条比率。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [metrics] weight_setting ● [metrics] 必需 ● [metrics] weight_multiplier

配置选项 = 默认值	类型	Description
weight_setting = []	列表值	<p>指标到权重修饰符的映射。</p> <p>此设置指定要权衡的指标以及每个指标的相对比率。这应该是一个字符串值，由一个或多个 <i>name=ratio</i> 对组成，用逗号分开，其中 name 是要权衡的指标的名称，而 ratio 是该指标的相对权重。</p> <p>请注意，如果比率设置为 0，指标值会被忽略，而是将 weight 设置为 [metrics] weight_of_unavailable 选项的值。</p> <p>例如，让我们考虑将此选项设置为以下情况：</p> <pre>name1=1.0, name2=-1.3`</pre> <p>最后的权重将是：</p> <pre>(name1.value * 1.0) + (name2.value * -1.3)`</pre> <p>可能的值：</p> <ul style="list-style-type: none"> 零个或多个键/值对列表，其中键是表示指标名称的字符串，值是该指标的数字权重。如果任何值设为 0，则忽略该值，权重将设置为 [metrics] weight_of_unavailable 选项的值。 <p>相关选项：</p> <ul style="list-style-type: none"> [metrics] weight_of_unavailable

9.1.28. mks

下表概述了 `/etc/nova/nova.conf` 文件中的 **[mks]** 组下可用的选项。

表 9.27. mks

配置选项 = 默认值	类型	Description
enabled = False	布尔值	为虚拟机启用图形控制台访问。

配置选项 = 默认值	类型	Description
mksproxy_base_url = http://127.0.0.1:6090/	URI 值	<p>MKS Web 控制台代理的位置</p> <p>响应中的 URL 指向 WebMKS 代理，该代理在客户端和实例运行的对应 vCenter 服务器之间启动代理。要使用基于 Web 的控制台访问，应该安装并配置 WebMKS 代理</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 必须是以下格式的有效 URL： http://host:port/ 或 https://host:port/

9.1.29. neutron

下表概述了 `/etc/nova/nova.conf` 文件中的 `[neutron]` 组下可用的选项。

表 9.28. neutron

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
connect-retries = None	整数值	应该尝试连接错误的最大重试次数。
connect-retry-delay = None	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。

配置选项 = 默认值	类型	Description
default_floating_pool = nova	字符串值	浮动 IP 池的默认名称。 指定用于分配浮动 IP 的浮动 IP 池的名称。只有在 Neutron 没有在端口绑定 responses 中指定浮动 IP 池名称时，才会使用此选项。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
endpoint-override = None	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version , min-version , 和/或 max-version 选项。
extension_sync_interval = 600	整数值	整数值，代表在查询 Neutron 扩展前要等待的秒数。下一次 Neutron 中需要创建资源的时间（以秒为单位）后，它会为已加载的扩展重新查询 Neutron。将值设为 0 将刷新没有等待的扩展。
http_retries = 3	整数值	neutronclient 在任何失败的 http 调用中应重试的次数。 0 表示只尝试一次连接。把它设置为任何正整数表示在失败连接中重试多次，例如将其设置为 3 表示连接总数为 4。 可能的值： <ul style="list-style-type: none">任何整数值。0 表示只尝试一次连接
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
<code>`metadata_proxy_shared_secret = `</code>	字符串值	此选项包含用于验证对 Neutron 元数据请求的代理请求的共享 secret 字符串。要使用，请求中必须提供 <i>X-Metadata-Provider-Signature</i> 标头。 相关选项： <ul style="list-style-type: none">service_metadata_proxy

配置选项 = 默认值	类型	Description
ovs_bridge = br-int	字符串值	Open vSwitch 集成网桥的默认名称。 指定 OpenvSwitch 使用的集成网桥接口名称。只有在 Neutron 没有在端口绑定响应中指定 OVS 网桥名称时，才会使用此选项。
password = None	字符串值	用户的密码
physnets = []	列表值	<p>此主机上存在 physnets 列表。</p> <p>对于列出的每个 physnet，会将额外部分 [neutron_physnet_\$PHYSNET] 添加到配置文件中。每个部分都必须配置有一个配置选项 numa_nodes，它应该是所有与 physnet 关联的所有 NUMA 节点的节点 ID 列表。例如：</p> <pre>[neutron] physnets = foo, bar [neutron_physnet_foo] numa_nodes = 0 [neutron_physnet_bar] numa_nodes = 0,1</pre> <p>任何未列出使用这个选项的 physnet 将被视为没有特定的 NUMA 节点关联性。</p> <p>无法以这种方式考虑 Tunnelled 网络(VXLAN、GRE、...)，而是使用 [neutron_tunnel] 组进行配置。例如：</p> <pre>[neutron_tunnel] numa_nodes = 1</pre> <p>相关选项：</p> <ul style="list-style-type: none"> ● [neutron_tunnel] numa_nodes 可用于为所有隧道网络配置 NUMA 关联性 ● [neutron_physnet_\$PHYSNET] numa_nodes 需要为由这个选项指定的每个 \$PHYSNET 值进度设置
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名

配置选项 = 默认值	类型	Description
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = network	字符串值	端点 URL 发现的默认 service_type。
service_metadata_proxy = False	布尔值	<p>当设置为 True 时，此选项表示 Neutron 将用于代理元数据请求并解析实例 ID。否则，实例 ID 必须在 X-Instance-ID 标头中传递给元数据请求。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● metadata_proxy_shared_secret
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名

配置选项 = 默认值	类型	Description
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。

9.1.30. notifications

下表概述了 `/etc/nova/nova.conf` 文件中的 `[notifications]` 组下可用的选项。

表 9.29. notifications

配置选项 = 默认值	类型	Description
bdms_in_notifications = False	布尔值	如果启用，请在版本通知有效负载中包含块设备信息。默认情况下，发送块设备信息会被禁用，因为提供这些信息可能会给系统造成一些开销，因为这些信息可能需要从数据库加载。
default_level = INFO	字符串值	传出通知的默认通知级别。
notification_format = unversioned	字符串值	<p>指定 nova 应发出哪些通知格式。</p> <p>版本化通知接口的功能与旧接口相同，版本化接口正在积极开发，因此新用户应使用版本化的接口。</p> <p>但是，传统的接口被 ceilometer 和其他成熟的 OpenStack 组件使用，因此它仍然是默认的。</p> <p>请注意，可以通过在 <code>[oslo_messaging_notifications]</code> 组中设置 <code>driver=noop</code> 来完全禁用通知。</p> <p>版本通知列表在 https://docs.openstack.org/nova/latest/reference/notifications.html 中看到</p>
notify_on_state_change = None	字符串值	<p>如果设置，在实例状态上发送 <code>compute.instance.update</code> 通知会改变。</p> <p>有关通知的详情，请参考 https://docs.openstack.org/nova/latest/reference/notifications.html。</p>

配置选项 = 默认值	类型	Description
versioned_notifications_topics = ['versioned_notifications']	列表值	<p>指定 nova 发布的版本通知的主题。</p> <p>对于大多数部署，默认值是正常的，很少需要更改。但是，如果您有一个使用版本通知的第三方服务，可能需要获得该服务的一个主题。Nova 将向此列表中的每个主题队列发送一条包含版本控制通知有效负载的消息。</p> <p>版本通知列表在 https://docs.openstack.org/nova/latest/reference/notifications.html中看到</p>

9.1.31. oslo_concurrency

下表概述了 `/etc/nova/nova.conf` 文件中的 `[oslo_concurrency]` 组下可用的选项。

表 9.30. oslo_concurrency

配置选项 = 默认值	类型	Description
disable_process_locking = False	布尔值	启用或禁用进程间锁定。
lock_path = None	字符串值	用于锁定文件的目录。为安全起见，指定目录应只由运行需要锁定的进程的用户写入。默认为环境变量 <code>OSLO_LOCK_PATH</code> 。如果使用外部锁定，则必须设置锁定路径。

9.1.32. oslo_messaging_amqp

下表概述了 `/etc/nova/nova.conf` 文件中的 `[oslo_messaging_amqp]` 组下可用的选项。

表 9.31. oslo_messaging_amqp

配置选项 = 默认值	类型	Description
addressing_mode = dynamic	字符串值	表示驱动程序使用的寻址模式。允许的值： <code>legacy</code> - 使用旧的不可路由寻址； <code>routable</code> - 使用可路由地址； <code>dynamic</code> - 如果消息总线不支持路由，使用 <code>legacy</code> ，如果支持，使用 <code>routable</code> 。
anycast_address = anycast	字符串值	在发送到一组消费者时附加到地址前缀。消息总线用来识别应在消费者之间以轮循方式发送的消息。
broadcast_prefix = broadcast	字符串值	广播到所有服务器时使用的地址前缀

配置选项 = 默认值	类型	Description
connection_retry_backoff = 2	整数值	每次失败故障转移尝试后，将 connection_retry_interval 增加到这秒数。
connection_retry_interval = 1	整数值	尝试重新连接前暂停的秒数。
connection_retry_interval_max = 30	整数值	connection_retry_interval + connection_retry_backoff 的最大限制
container_name = None	字符串值	AMQP 容器的名称必须全局唯一。默认为生成的 UUID
default_notification_exchange = None	字符串值	通知地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_notification_exchange if set else control_exchange if set else <i>notify</i>
default_notify_timeout = 30	整数值	发送通知消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_reply_retry = 0	整数值	重新发送回复消息的最大尝试次数，因为可恢复错误而失败。
default_reply_timeout = 30	整数值	rpc 回复消息发送的截止时间。
default_rpc_exchange = None	字符串值	RPC 地址中使用的交换名称。Exchange name resolution priority: Target.exchange if set default_rpc_exchange if set other control_exchange if set else <i>rpc</i>
default_send_timeout = 30	整数值	rpc cast 或调用消息发送的截止时间。仅在调用者不提供超时到期时使用。
default_sender_link_timeout = 600	整数值	调度闲置发送者链接清除的持续时间。过期后分离链接。
group_request_prefix = unicast	字符串值	发送到组中任何服务器时的地址前缀
idle_timeout = 0	整数值	不活跃连接的超时时间（以秒为单位）
link_retry_delay = 10	整数值	重新连接 AMQP 1.0 链接之间的暂停时间，该链接因为可恢复的错误而失败。
multicast_address = multicast	字符串值	在发送异常消息时，附加到地址前缀。消息总线用来识别 fanout 消息。

配置选项 = 默认值	类型	Description
notify_address_prefix = openstack.org/om/notify	字符串值	所有生成的通知地址的地址前缀
notify_server_credit = 100	整数值	传入通知消息的窗口大小
pre_settled = ['rpc-cast', 'rpc-reply']	多值	发送此类型的消息预先组装。预先设置的消息不会接收来自 peer 的确认信息。请注意：如果发送失败，则预先设置的消息可能会被静默丢弃。允许的值： <i>rpc-call</i> - 发送 RPC Calls <i>pre-settled</i> <i>rpc-reply</i> - 发送 RPC Replies <i>pre-settled</i> <i>rpc-cast</i> - Send RPC Casts <i>pre-settled</i> <i>notify</i> - Send Notifications <i>pre-settled</i>
pseudo_vhost = True	布尔值	为不支持虚拟主机的消息总线启用虚拟主机支持（如 qpidd）。当设置为 true 时，虚拟主机名将添加到所有消息总线地址中，从而有效地为每个虚拟主机 <i>创建专用子网</i> 。如果消息总线支持使用 AMQP 1.0 Open performative 中的 <i>hostname</i> 字段作为虚拟主机的名称，设置为 False。
reply_link_credit = 200	整数值	传入 RPC 消息的窗口大小。
rpc_address_prefix = openstack.org/om/rpc	字符串值	所有生成的 RPC 地址的地址前缀
rpc_server_credit = 100	整数值	传入 RPC Request 消息的窗口大小
<code>`sas_config_dir = `</code>	字符串值	包含 SASL 配置的目录路径
<code>`sas_config_name = `</code>	字符串值	配置文件名称（不带 .conf 后缀）
<code>`sas_default_realm = `</code>	字符串值	如果用户名中没有 realm，则要使用的 SASL 域
<code>`sas_mechanisms = `</code>	字符串值	可接受的 SASL 机制的以空格分隔的列表
server_request_prefix = exclusive	字符串值	发送到特定服务器时使用的地址前缀
ssl = False	布尔值	尝试通过 SSL 连接。如果没有给出其他与 ssl 相关的参数，它将使用系统的 CA-bundle 来验证服务器证书。
<code>`ssl_ca_file = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_cert_file = `</code>	字符串值	自我标识用于客户端身份验证的证书 PEM 文件
<code>`ssl_key_file = `</code>	字符串值	用于为 <code>ssl_cert_file</code> 证书签名的私钥 PEM 文件（可选）

配置选项 = 默认值	类型	Description
ssl_key_password = None	字符串值	用于解密 ssl_key_file 的密码（如果加密）
ssl_verify_vhost = False	布尔值	默认情况下，SSL 检查服务器证书中的名称与 transport_url 中的主机名匹配。在某些配置中，最好使用虚拟主机名。例如，如果服务器使用 Server Name Indication TLS extension (rfc6066) 来为每个虚拟主机提供证书。如果服务器的 SSL 证书使用虚拟主机名称而不是 DNS 名称，则将 ssl_verify_vhost 设置为 True。
trace = False	布尔值	debug : 将 AMQP 帧转储到 stdout
unicast_address = unicast	字符串值	在发送到特定 RPC/Notification 服务器时，附加到地址前缀。消息总线用来识别发送到单个目的地的消息。

9.1.33. oslo_messaging_kafka

下表概述了 /etc/nova/nova.conf 文件中的 [oslo_messaging_kafka] 组下可用的选项。

表 9.32. oslo_messaging_kafka

配置选项 = 默认值	类型	Description
compression_codec = none	字符串值	producer 生成的所有数据的压缩 codec。如果没有设置，则不会使用压缩。请注意，这允许的值取决于 kafka 版本
conn_pool_min_size = 2	整数值	连接过期策略的池大小限制
conn_pool_ttl = 1200	整数值	池中闲置连接的时间延迟
consumer_group = oslo_messaging_consumer	字符串值	Kafka 消费者的组 ID。一个组中的消费者将协调消息消耗
enable_auto_commit = False	布尔值	启用异步消费者提交
kafka_consumer_timeout = 1.0	浮点值	Kafka 用户的默认超时
kafka_max_fetch_bytes = 1048576	整数值	Kafka consumer 的最大获取字节数

配置选项 = 默认值	类型	Description
max_poll_records = 500	整数值	轮询调用中返回的最大记录数
pool_size = 10	整数值	Kafka Consumers 的池大小
producer_batch_size = 16384	整数值	producer async 发送的批处理大小
producer_batch_timeout = 0.0	浮点值	以秒为单位的 KafkaProducer 批处理延迟上限
sasl_mechanism = PLAIN	字符串值	当安全协议是 SASL 时的机制
security_protocol = PLAINTEXT	字符串值	用于与代理通信的协议
<code>`ssl_cafile = `</code>	字符串值	用于验证服务器证书的 CA 证书 PEM 文件
<code>`ssl_client_cert_file = `</code>	字符串值	用于身份验证的客户端证书 PEM 文件。
<code>`ssl_client_key_file = `</code>	字符串值	用于身份验证的客户端密钥 PEM 文件。
<code>`ssl_client_key_password = `</code>	字符串值	用于身份验证的客户端密钥密码文件。

9.1.34. oslo_messaging_notifications

下表概述了 `/etc/nova/nova.conf` 文件中的 `[oslo_messaging_notifications]` 组下可用的选项。

表 9.33. oslo_messaging_notifications

配置选项 = 默认值	类型	Description
driver = []	多值	处理发送通知的驱动程序。可能的值有 messaging, messagingv2, routing, log, test, noop
retry = -1	整数值	重新发送因为可恢复错误而发送失败的通知消息的最大尝试次数。0 - 不重试, -1 - indefinite
topics = ['notifications']	列表值	用于 OpenStack 通知的 AMQP 主题。
transport_url = None	字符串值	代表用于通知的消息传递驱动程序的 URL。如果没有设置, 我们回退到用于 RPC 的同一配置。

9.1.35. oslo_messaging_rabbit

下表概述了 `/etc/nova/nova.conf` 文件中的 `[oslo_messaging_rabbit]` 组下可用的选项。

表 9.34. oslo_messaging_rabbit

配置选项 = 默认值	类型	Description
amqp_auto_delete = False	布尔值	AMQP 中的自动删除队列。
amqp_durable_queues = False	布尔值	在 AMQP 中使用持久队列。
direct_mandatory_flag = True	布尔值	(DEPRECATED)为直接发送启用/禁用 RabbitMQ 强制标记。直接发送用作回复，因此在客户端队列不存在时引发 MessageUndeliverable 异常。MessageUndeliverable 异常将用于循环使发送者恢复。此标志已弃用，且无法停用此功能
enable_cancel_on_failover = False	布尔值	启用 x-cancel-on-ha-failover 标志，使 rabbitmq 服务器将取消并通知 consumerswhen 队列停机
heartbeat_in_pthread = False	布尔值	默认情况下，通过原生 python 线程运行健康检查心跳线程。如果此选项等于 False，则健康检查 heartbeat 将继承父进程的执行模型。例如，如果父进程使用 eventlet/greenlet 对 stdlib 进行补丁，则心跳将通过绿色线程运行。这个选项应该只针对 wsgi 服务设置为 True。
heartbeat_rate = 2	整数值	heartbeat_timeout_threshold 期间的我们检查心跳的频率。
heartbeat_timeout_threshold = 60	整数值	如果 heartbeat 的 keep-alive 失败(0 禁用 heartbeat)，则 Rabbit 代理被视为 down 的秒数。
kombu_compression = None	字符串值	EXPERIMENTAL : Possible 值为 : gzip、bz2。如果没有设置压缩，则不会使用。这个选项可能在以后的版本中不可用。
kombu_failover_strategy = round-robin	字符串值	确定当当前连接到的 RabbitMQ 节点不可用时如何选择下一个 RabbitMQ 节点。仅在配置中提供多个 RabbitMQ 节点时生效。
kombu_missing_consumer_retry_timeout = 60	整数值	在放弃发送其回复前等待缺少的客户端的时间。这个值不应比 rpc_response_timeout 长。
kombu_reconnect_delay = 1.0	浮点值	响应 AMQP 消费者取消通知前等待多久。

配置选项 = 默认值	类型	Description
rabbit_ha_queues = False	布尔值	尝试在 RabbitMQ 中使用 HA 队列(x-ha-policy: all)。如果更改这个选项，您必须擦除 RabbitMQ 数据库。在 RabbitMQ 3.0 中，在声明队列时，x-ha-policy 参数不再控制队列镜像。如果您只想确保所有队列（带有自动生成的名称的队列除外）都在所有节点上进行镜像，请运行： <code>"rabbitmqctl set_policy HA ^(?!amq\.) gem {"ha-mode": "all"}"</code>
rabbit_interval_max = 30	整数值	RabbitMQ 连接重试的最大间隔。默认值为 30 秒。
rabbit_login_method = AMQPLAIN	字符串值	RabbitMQ 登录方法。
rabbit_qos_prefetch_count = 0	整数值	指定要预抓取的消息数量。设置为零可允许无限消息。
rabbit_retry_backoff = 2	整数值	连接到 RabbitMQ 时重试之间的 backoff 时长。
rabbit_retry_interval = 1	整数值	使用 RabbitMQ 重试连接的频率。
rabbit_transient_queues_ttl = 1800	整数值	正整数代表队列 TTL (x-expires)的持续时间（以秒为单位）。TTL 期间未使用的队列将被自动删除。参数仅影响 reply 和 fanout 队列。
ssl = False	布尔值	通过 SSL 连接。
<code>`ssl_ca_file = `</code>	字符串值	SSL 证书颁发机构文件（仅在启用了 SSL 时评估）。
<code>`ssl_cert_file = `</code>	字符串值	SSL 证书文件（仅在启用了 SSL 时评估）。
<code>`ssl_key_file = `</code>	字符串值	SSL 密钥文件（仅在启用了 SSL 时valid）。
<code>`ssl_version = `</code>	字符串值	要使用的 SSL 版本（仅在启用了 SSL 时评估）。有效值为 TLSv1 和 SSLv23。SSLv2、SSLv3、TLSv1_1 和 TLSv1_2 可能在某些发行版中提供。

9.1.36. oslo_middlewares

下表概述了 `/etc/nova/nova.conf` 文件中的 `[oslo_middlewares]` 组下可用的选项。

表 9.35. oslo_middlewares

配置选项 = 默认值	类型	Description
enable_proxy_headers_parsing = False	布尔值	应用程序是否在代理后面。这决定了中间件是否应该解析标头。

配置选项 = 默认值	类型	Description
max_request_body_size = 114688	整数值	每个请求的最大正文大小，以字节为单位。
secure_proxy_ssl_header = X-Forwarded-Proto	字符串值	用于确定原始请求协议方案的 HTTP 标头，即使它被 SSL 终止代理隐藏也是如此。

9.1.37. oslo_policy

下表概述了 `/etc/nova/nova.conf` 文件中的 `[oslo_policy]` 组下可用的选项。

表 9.36. oslo_policy

配置选项 = 默认值	类型	描述
enforce_new_defaults = False	布尔值	这个选项控制在评估策略时是否使用旧的弃用默认值。如果为 True ，则不会评估旧的已弃用的默认值。这意味着，如果旧默认值允许任何现有令牌，但不允许使用新默认值，它将被禁止。我们鼓励启用此标志以及 enforce_scope 标志，以便您可以同时获得新默认值和 scope_type 的好处
enforce_scope = False	布尔值	此选项控制在评估策略时是否强制执行范围。如果为 True ，则请求中使用的令牌范围将与被强制执行策略的 scope_types 进行比较。如果范围不匹配，则引发 InvalidScope 异常。如果为 False ，则会记录一条信息，告知操作员使用了不匹配的范围调用策略。
policy_default_rule = default	字符串值	默认规则。未找到请求的规则时强制执行。
policy_dirs = ['policy.d']	多值	存储策略配置文件的目录。它们可以相对于 <code>config_dir</code> 选项或绝对路径定义的搜索路径中的任何目录。由 <code>policy_file</code> 定义的文件必须存在，才能搜索这些目录。缺少或空目录将被忽略。
policy_file = policy.yaml	字符串值	将角色映射到给定服务权限的文件的相对或绝对路径。必须使用与配置文件设置相关的相对路径。
remote_content_type = application/x-www-form-urlencoded	字符串值	用于发送和接收基于 REST 策略检查的数据的内容类型
remote_ssl_ca_cert_file = None	字符串值	基于 REST 的策略检查的 ca cert 文件的绝对路径
remote_ssl_client_cert_file = None	字符串值	基于 REST 的策略检查的客户端证书的绝对路径

配置选项 = 默认值	类型	描述
<code>remote_ssl_client_key_file = None</code>	字符串值	绝对路径客户端密钥文件 REST 基于策略检查
<code>remote_ssl_verify_server_cert = False</code>	布尔值	基于 REST 的策略检查的服务器身份验证

9.1.38. pci

下表概述了 `/etc/nova/nova.conf` 文件中的 `[pci]` 组下可用的选项。

表 9.37. pci

配置选项 = 默认值	类型	Description
<code>alias = []</code>	多值	<p>PCI 透传设备要求的别名。</p> <p>这允许用户在额外规格中指定类别的别名，而无需重复所有 PCI 属性要求。</p> <p>这应该为 nova-api 服务配置，并假定您要为每个 nova-compute 服务使用 <code>move</code> 操作。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 描述别名的 JSON 值的字典。例如： <pre>alias = { "name": "QuickAssist", "product_id": "0443", "vendor_id": "8086", "device_type": "type-PCI", "numa_policy": "required" }</pre> <p>This defines an alias for the Intel QuickAssist card. (multi valued). Valid key values are :</p> <pre>`name` Name of the PCI alias.</pre> <pre>`product_id` Product ID of the device in hexadecimal.</pre> <pre>`vendor_id` Vendor ID of the device in hexadecimal.</pre>

配置选项 = 默认值	类型	Description
		<p><code>`device_type`</code> Type of PCI device. Valid values are: <code>`type-PCI`</code>, <code>`type-PF`</code> and <code>`type-VF`</code>. Note that <code>`"device_type": "type-PF"`</code> must be specified if you wish to passthrough a device that supports SR-IOV in its entirety.</p> <p><code>`numa_policy`</code> Required NUMA affinity of device. Valid values are: <code>`legacy`</code>, <code>`preferred`</code> and <code>`required`</code>.</p> <ul style="list-style-type: none"> 通过重复选项（而不是通过指定）来支持多个别名 <p>列表值)</p> <pre>alias = { "name": "QuickAssist-1", "product_id": "0443", "vendor_id": "8086", "device_type": "type-PCI", "numa_policy": "required" } alias = { "name": "QuickAssist-2", "product_id": "0444", "vendor_id": "8086", "device_type": "type-PCI", "numa_policy": "required" }</pre>
<p><code>passthrough_whitelist = []</code></p>	<p>多值</p>	<p>可用于虚拟机的 PCI 设备列表。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 描述白名单 PCI 设备的 JSON 字典。它应该采用 <p>以下格式</p> <pre>["vendor_id": "<id>", ["product_id": " <id>", ["address": "[[[[<domain>]:] <bus>]:[<slot>][.<function>]]" "devname": "<name>",] { "tag": " <tag_value>", }</pre> <p>Where <code>`[`</code> indicates zero or one occurrences, <code>`{`</code> indicates zero or multiple occurrences, and <code>`&verbar;`</code> mutually exclusive options. Note that any missing fields are automatically wildcarded.</p> <p>Valid key values are :</p> <p><code>`vendor_id`</code> Vendor ID of the device in hexadecimal.</p>

配置选项 = 默认值	类型	Description
		<p><code>`product_id`</code> Product ID of the device in hexadecimal.</p> <p><code>`address`</code> PCI address of the device. Both traditional glob style and regular expression syntax is supported. Please note that the address fields are restricted to the following maximum values:</p> <ul style="list-style-type: none"> ● 域 - 0xFFFF ● 总线 - 0xFF ● slot - 0x1F ● function - 0x7 <p><code>`devname`</code> Device name of the device (for e.g. interface name). Not all PCI devices have a name.</p> <p><code>`<tag>`</code> Additional <code>`<tag>`</code> and <code>`<tag_value>`</code> used for matching PCI devices. Supported <code>`<tag>`</code> values are :</p> <ul style="list-style-type: none"> ○ physical_network ○ trusted <p>有效示例为</p> <pre> passthrough_whitelist = {"devname":"eth0", "physical_network":"physnet"} passthrough_whitelist = {"address":"0a:00."} passthrough_whitelist = {"address":"0a:00.", "physical_network":"physnet1"} passthrough_whitelist = {"vendor_id":"1137", "product_id":"0071"} passthrough_whitelist = {"vendor_id":"1137", "product_id":"0071", "address": "0000:0a:00.1", "physical_network":"physnet1"} passthrough_whitelist = {"address": {"domain": ".", "bus": "02", "slot": "01", "function": "[2-7]"}, "physical_network":"physnet1"} passthrough_whitelist = </pre>

配置选项 = 默认值	类型	Description
		<pre> {"address":{"domain": ".", "bus": "02", "slot": "0[1-2]", "function": "*"}, "physical_network": "physnet1"} passthrough_whitelist = {"devname": "eth0", "physical_network": "physnet1", "trusted": "true"} </pre> <p>以下无效，因为它们指定了相互排斥的选项</p> <pre> passthrough_whitelist = {"devname": "eth0", "physical_network": "physnet", "address": "0a:00."} </pre> <ul style="list-style-type: none"> 与上述格式对应的 JSON 字典列表。对于 <p>示例</p> <pre> passthrough_whitelist = [{"product_id": "0001", "vendor_id": "8086"}, {"product_id": "0002", "vendor_id": "8086"}] </pre>

9.1.39. placement

下表概述了 `/etc/nova/nova.conf` 文件中的 `[placement]` 组下可用的选项。

表 9.38. placement

配置选项 = 默认值	类型	Description
<code>auth-url = None</code>	字符串值	身份验证 URL
<code>auth_section = None</code>	字符串值	从中加载插件特定选项的配置部分
<code>auth_type = None</code>	字符串值	要加载的身份验证类型
<code>cafile = None</code>	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
<code>certFile = None</code>	字符串值	PEM 编码的客户端证书证书文件
<code>collect-timing = False</code>	布尔值	收集每个 API 调用计时信息。
<code>connect-retries = None</code>	整数值	应该尝试连接错误的最大重试次数。
<code>connect-retry-delay = None</code>	浮点值	两次重试之间延迟（以秒为单位）用于连接错误。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
<code>default-domain-id = None</code>	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。

配置选项 = 默认值	类型	Description
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
endpoint-override = None	字符串值	始终将此端点 URL 用于此客户端的请求。注意：应在此处指定未版本化的端点；若要请求特定的 API 版本，使用 version , min-version , 和/或 max-version 选项。
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
region-name = None	字符串值	用于端点 URL 发现的默认 region_name。
service-name = None	字符串值	用于端点 URL 发现的默认 service_name。
service-type = placement	字符串值	端点 URL 发现的默认 service_type。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
status-code-retries = None	整数值	应该尝试重新分配 HTTP 状态代码的最大重试次数。
status-code-retry-delay = None	浮点值	在两次重试之间延迟（以秒为单位）用于可重新分配状态代码。如果没有设置，则以 0.5 秒开始的指数重试，最多使用 60 秒。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID

配置选项 = 默认值	类型	Description
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名
valid-interfaces = ['internal', 'public']	列表值	首选接口列表，用于端点 URL。

9.1.40. powervm

下表概述了 `/etc/nova/nova.conf` 文件中的 `[powervm]` 组下可用的选项。

表 9.39. powervm

配置选项 = 默认值	类型	Description
disk_driver = localdisk	字符串值	用于 PowerVM 磁盘的磁盘驱动程序。PowerVM 支持 localdisk 和 PowerVM 共享存储池磁盘驱动程序。 相关选项： <ul style="list-style-type: none"> ● volume_group_name - 使用 localdisk 时需要
proc_units_factor = 0.1	浮点值	用于计算提供给每个 vCPU 的物理处理器计算能力的因素。例如值 1.0 表示整个物理处理器，而 0.05 代表物理处理器的 1/20。
<code>volume_group_name = `</code>	字符串值	用于块设备操作的卷组。如果 disk_driver 是 localdisk，则必须指定此属性。强烈建议您不要使用 rootvg，因为管理分区使用并填充它将导致失败。

9.1.41. privsep

下表概述了 `/etc/nova/nova.conf` 文件中的 `[privsep]` 组下可用的选项。

表 9.40. privsep

配置选项 = 默认值	类型	Description
capabilities = []	列表值	privsep 守护进程保留的 Linux 功能列表。
group = None	字符串值	privsep 守护进程应作为。
helper_command = None	字符串值	如果不使用"fork"方法，调用以启动 privsep 守护进程。如果没有指定，则使用"sudo privsep-helper"和用于重新创建当前配置的参数生成默认值。此命令必须接受合适的 --privsep_context 和 --privsep_sock_path 参数。
logger_name = oslo_privsep.daemon	字符串值	用于此 privsep 上下文的日志记录器名称。默认情况下，带有 oslo_privsep.daemon 的所有上下文日志。
thread_pool_size = <based on operating system>	整数值	可用于 privsep 的线程数量，以同时运行进程。默认为系统中的 CPU 内核数。
user = None	字符串值	privsep 守护进程应以。

9.1.42. profiler

下表概述了 `/etc/nova/nova.conf` 文件中的 `[profiler]` 组下可用的选项。

表 9.41. profiler

配置选项 = 默认值	类型	Description
connection_string = messaging://	字符串值	<p>通知程序后端的连接字符串。</p> <p>默认值为 messaging://，它将通知程序设置为 oslo_messaging。</p> <p>可能的值示例：</p> <ul style="list-style-type: none"> ● messaging:// - 使用 oslo_messaging 驱动程序来发送范围。 ● redis://127.0.0.1:6379 - 使用 redis 驱动程序发送 span。 ● MongoDB://127.0.0.1:27017 - 使用 mongodb 驱动程序进行发送范围。 ● Elasticsearch://127.0.0.1:9200 - 使用 elasticsearch 驱动程序来发送 span。 ● Jaeger://127.0.0.1:6831 - 使用 jaeger tracing 作为发送 span 的驱动程序。

配置选项 = 默认值	类型	Description
enabled = False	布尔值	<p>为此节点上的所有服务启用性能分析。</p> <p>默认值为 False（完全禁用性能分析功能）。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用该功能 ● false：禁用该功能。分析无法通过此项目操作启动。如果性能分析由另一个项目触发，则此项目部分将为空。
es_doc_type = notification	字符串值	elasticsearch 中通知索引的文档类型。
es_scroll_size = 10000	整数值	Elasticsearch 批量分割大型请求。此参数定义每个批处理的最大大小（例如：es_scroll_size=10000）。
es_scroll_time = 2m	字符串值	此参数是一个时间值参数（例如：es_scroll_time=2m），指示参与搜索的节点将维护相关资源以继续和支持它。
filter_error_trace = False	布尔值	<p>启用包含 error/exception 的过滤器追踪来分隔的位置。</p> <p>默认值为 False。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用包含 error/exception 的过滤器追踪。 ● false：禁用过滤器。
hmac_keys = SECRET_KEY	字符串值	<p>用于加密上下文数据的 secret 密钥，以用于性能分析。</p> <p>此字符串值应该具有以下格式：<key1>[,<key2>],...<keyn>]，其中每个键都是随机字符串。通过 REST API 触发性能分析的用户必须在 REST API 调用的标头中设置其中一个键，使其包含此节点的性能分析结果。</p> <p>"enabled"标志和"hmac_keys"配置选项都应设置为启用性能分析。此外，为了跨所有服务生成正确的性能分析信息，OpenStack 项目之间至少需要一个关键。这样可确保它可用于客户端生成 trace，其中包含所有可能的资源的信息。</p>
sentinel_service_name = mymaster	字符串值	<p>Redis sentinel 使用服务名称来识别 master redis 服务。此参数定义名称（例如：sentinal_service_name=mymaster）。</p>

配置选项 = 默认值	类型	Description
socket_timeout = 0.1	浮点值	Redis Sentinel 在连接上提供了一个 timeout 选项。此参数定义该超时（例如：socket_timeout=0.1）。
trace_sqlalchemy = False	布尔值	<p>在服务中启用 SQL 请求性能分析。</p> <p>默认值为 False (SQL 请求不会被跟踪)。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：启用 SQL 请求性能分析。每个 SQL 查询都将是 trace 的一部分，可以通过花费的时间来分析。 ● false：禁用 SQL 请求性能分析。花费的时间仅显示在更高级别的操作上。无法以这种方式分析单个 SQL 查询。

9.1.43. quota

下表概述了 `/etc/nova/nova.conf` 文件中的 **[quota]** 组下可用的选项。

表 9.42. quota

配置选项 = 默认值	类型	Description
cores = 20	整数值	<p>每个项目允许的实例内核数或 vCPU 数量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数或 0。 ● -1 禁用配额。

配置选项 = 默认值	类型	Description
count_usage_from_placement = False	布尔值	<p>启用放置服务的配额使用量计数。</p> <p>从 Train 开始，可以计算内核的配额使用量，以及来自放置服务和来自 API 数据库的实例，而不是从单元数据库计数。</p> <p>如果每个放置部署只有一个 Nova 部署运行，这可以正常工作。但是，如果操作器运行多个 Nova 部署共享放置部署，则不应将此选项设置为 True，因为当前放置服务没有每个 Nova 部署分区资源提供程序的方法。当此选项保留为默认值或设置为 False 时，Nova 将使用传统计数方法来计算其单元数据库中的实例、核心和 RAM 的配额使用量。</p> <p>请注意，如果此选项设置为 True，则与调整大小相关的配额使用量行为将会受到影响。在调整大小期间在源上保存分配时，会在目标上声明放置资源分配，直到调整大小确认或恢复为止。在这段时间中，当服务器处于 VERIFY_RESIZE 状态时，配额使用量将反映源和目的地上的资源消耗。这很有用，因为它保留了 downsize 恢复空间，但也意味着配额使用量将在确认或恢复调整大小前被忽略。</p> <p>对于处于 ERROR 状态的未调度的服务器，行为也会不同。永不调度到计算主机的服务器没有放置分配，因此它不会消耗内核和 RAM 的配额使用量。</p> <p>对于处于 SHELVED_OFFLOADED 状态的服务器，行为会有所不同。SHELVED_OFFLOADED 状态的服务器没有放置分配，因此它不会消耗内核和 RAM 的配额使用量。请注意，由于此，如果用户没有足够的配额来支持服务器所需的内核和 RAM，则可能会拒绝服务器的请求。</p> <p>在从放置计算使用前，必须完成 populate_queued_for_delete 和 populate_user_id 在线数据迁移。在数据迁移完成前，如果此配置选项被设置为 True，则系统将回退到来自单元数据库的旧配额使用量计数，具体取决于每个配额检查期间 EXISTS 数据库查询的结果。希望避免从 EXISTS 查询性能达到的 Operator 应该等待将此配置选项设置为 True，直到它们通过 nova-manage db online_data_migrations 完成在线数据迁移。</p>
driver = nova.quota.DbQuotaDriver	字符串值	<p>为配额检查提供抽象。用户可以配置特定的驱动程序，以用于配额检查。</p>

配置选项 = 默认值	类型	Description
injected_file_content_bytes = 10240	整数值	<p>每个注入的文件允许的字节数。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数或 0。 ● -1 禁用配额。
injected_file_path_length = 255	整数值	<p>允许的最大注入的文件路径长度。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数或 0。 ● -1 禁用配额。
injected_files = 5	整数值	<p>允许注入的文件数量。</p> <p>文件注入允许用户在启动时通过注入数据来自定义实例的个人。只允许文本文件注入：不接受二进制或 ZIP 文件。在文件注入过程中，任何匹配指定文件的现有文件都会被重命名为包括带有时间戳的 .bak 扩展。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数或 0。 ● -1 禁用配额。
instances = 10	整数值	<p>每个项目允许的实例数量。</p> <p>可能的值</p> <ul style="list-style-type: none"> ● 正整数或 0。 ● -1 禁用配额。
key_pairs = 100	整数值	<p>每个用户允许的最大密钥对数。</p> <p>用户可以为每个项目至少创建一个密钥对，并将密钥对用于属于该项目的多个实例。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数或 0。 ● -1 禁用配额。

配置选项 = 默认值	类型	Description
metadata_items = 128	整数值	<p>每个实例允许的元数据项数。</p> <p>用户可以在实例创建过程中将元数据与实例关联。此元数据采用键值对的形式。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数或 0。 ● -1 禁用配额。
ram = 51200	整数值	<p>每个项目允许的实例 RAM 数量。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数或 0。 ● -1 禁用配额。
recheck_quota = True	布尔值	<p>在创建资源后重新检查配额，以防止超过配额。</p> <p>默认值为 True（在资源创建后重新检查配额），但可以设置为 False，以避免在允许超过配额时避免额外的负载。例如，当用户设置为 False 时，如果用户进行高度并行的 REST API 请求来创建服务器，则它们可以创建比其允许的配额更多的服务器。如果配额是 10 个服务器，则它们可能会在突发期间创建 50。在突发后，他们将无法创建任何更多服务器，但它们能够在删除前保留其 50 个服务器。</p> <p>初始配额检查是在创建资源之前完成的，因此如果多个并行请求同时到达，则所有请求都可以通过配额检查并创建资源，可能超过配额。当 recheck_quota 为 True 时，在创建资源后会第二次检查配额，如果资源超过配额，它将被删除，并且会引发 OverQuota，通常是对 REST API 用户的 403 响应。这使得用户无法超过其配额，但可能会发现 REST API 用户可能会在有冲突的情况下拒绝 403 响应，即使用户在发出请求时也有足够的配额可用。</p>
server_group_members = 10	整数值	<p>每个服务器组的最大服务器数。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数或 0。 ● -1 禁用配额。

配置选项 = 默认值	类型	Description
server_groups = 10	整数值	<p>每个项目的最大服务器组数。</p> <p>服务器组用于控制一组服务器或实例的关联性和反关联性调度策略。减少配额不会影响任何现有组，但不允许使用新的服务器到已超过配额的组中。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 正整数或 0。 ● -1 禁用配额。

9.1.44. rdp

下表概述了 `/etc/nova/nova.conf` 文件中的 `[rdp]` 组下可用的选项。

表 9.43. rdp

配置选项 = 默认值	类型	Description
enabled = False	布尔值	<p>启用与远程桌面协议(RDP)相关功能。</p> <p>Hyper-V 与 Nova 计算节点上使用的大多数虚拟机监控程序不同，使用 RDP 而不是 VNC 和 SPICE 作为桌面共享协议来提供实例控制台访问。这个选项为 Hyper-V 创建的虚拟机启用 RDP 访问。</p> <p>注： RDP 应该只在支持 Hyper-V 虚拟化平台的计算节点上启用。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● compute_driver: 必须为超v。

配置选项 = 默认值	类型	Description
html5_proxy_base_url = http://127.0.0.1:6083/	URI 值	<p>最终用户用来连接到 RDP HTML5 控制台代理的 URL。使用这个 token-embedded URL 调用控制台代理服务，并建立到正确实例的连接。</p> <p>RDP HTML5 控制台代理服务需要配置为侦听此处配置的地址。通常，控制台代理服务将在控制器节点上运行。用作默认值的 localhost 地址只能在单一节点环境中工作，例如 devstack。</p> <p>RDP HTML5 代理允许用户使用 RDP 通过任何 Windows 服务器或工作站的 Web 或图形控制台访问。RDP HTML5 控制台代理服务包括 FreeRDP, wsgate。See https://github.com/FreeRDP/FreeRDP-WebConnect</p> <p>可能的值：</p> <ul style="list-style-type: none"> • <scheme>://<ip-address>:<port-number>/ <div style="border-left: 2px solid black; padding-left: 10px; margin-left: 20px;"> <p>The scheme must be identical to the scheme configured for the RDP HTML5 console proxy service. It is `http` or `https`.</p> </div> <div style="border-left: 2px solid black; padding-left: 10px; margin-left: 20px;"> <p>The IP address must be identical to the address on which the RDP HTML5 console proxy service is listening.</p> </div> <div style="border-left: 2px solid black; padding-left: 10px; margin-left: 20px;"> <p>The port must be identical to the port on which the RDP HTML5 console proxy service is listening.</p> </div> <p>相关选项：</p> <ul style="list-style-type: none"> • RDP.enabled : 必须为 html5_proxy_base_url 设置为 True。

9.1.45. remote_debug

下表概述了 `/etc/nova/nova.conf` 文件中的 `[remote_debug]` 组下可用的选项。

表 9.44. remote_debug

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
host = None	主机地址值	<p>要连接的调试主机(IP 或名称)。</p> <p>当您想通过不同主机上运行的调试器连接到 nova 服务时，会使用此命令行参数。</p> <p>请注意，使用远程 debug 选项更改 nova 如何使用 eventlet 库来支持 async IO。这可能会导致在正常操作下不会发生失败。自行使用风险。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 远程主机的 IP 地址作为命令行参数到 nova 服务。 <p>例如：</p> <pre>nova-compute --config-file /etc/nova/nova.conf --remote_debug- host <IP address of the debugger></pre>
port = None	端口值	<p>要连接的调试端口。</p> <p>此命令行参数允许您通过不同主机上运行的调试器指定您要用来连接到 nova 服务的端口。</p> <p>请注意，使用远程 debug 选项更改 nova 如何使用 eventlet 库来支持 async IO。这可能会导致在正常操作下不会发生失败。自行使用风险。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 您要用作 nova 服务的命令行参数的端口号。 <p>例如：</p> <pre>nova-compute --config-file /etc/nova/nova.conf --remote_debug- host <IP 地址 debugger> -- remote_debug-port <port debugger 正 在侦听>。</pre>

9.1.46. scheduler

下表概述了 `/etc/nova/nova.conf` 文件中的 `[scheduler]` 组下可用的选项。

表 9.45. scheduler

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
discover_hosts_in_cells_interval = -1	整数值	<p>定期任务间隔。</p> <p>这个值控制调度程序应尝试发现添加到单元中的新主机的频率（以秒为单位）。如果为负（默认），则不会发生自动发现。</p> <p>计算节点上线并经常需要启用的部署，在添加节点时可能希望手动发现主机，以避免持续检查开销。如果启用，每次运行时，我们将在每次运行时都会从每个单元数据库中选择任何未映射的主机。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 整数，其中整数对应于定期任务间隔（以秒为单位）。0 使用默认间隔(60 秒)。负值禁用定期任务。
enable_isolated_aggregate_filtering = False	布尔值	<p>将聚合的使用限制为具有匹配元数据的实例。</p> <p>此设置允许调度程序根据聚合元数据和实例 flavor/image 中匹配的所需特征来限制聚合中的主机。如果使用带有键特征的属性配置了聚合：trait:\$TRAIT_NAME 和值 required，则实例类别 extra_specs 和/或镜像元数据还必须包含 trait:\$TRAIT_NAME=required 有资格调度到该聚合中的主机。更多技术详情： https://docs.openstack.org/nova/latest/reference/isolate-aggregates.html</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 布尔值。
image_metadata_prefilter = False	布尔值	<p>使用 placement 根据镜像元数据过滤主机。</p> <p>此设置会导致调度程序将已知的镜像元数据属性转换为放置所需的特征，以根据镜像元数据过滤主机。此功能需要主机支持，且目前由以下计算驱动程序支持：</p> <ul style="list-style-type: none"> ● libvirt.LibvirtDriver（自 Ussuri (21.0.0) 开始） <p>可能的值：</p> <ul style="list-style-type: none"> ● 布尔值。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [compute] compute_driver

配置选项 = 默认值	类型	Description
limit_tenants_to_placement_aggregate = False	布尔值	<p>将租户限制为特定的放置聚合。</p> <p>此设置会导致调度程序使用设置为传入请求的项目的 filter_tenant_id 的元数据键查找主机聚合，并从放置请求请求结果限制为该聚合。通过将序列号附加到密钥，可以将多个租户添加到单个聚合中，如 filter_tenant_id:123。</p> <p>必须在放置中镜像匹配的聚合 UUID，才能正常工作。如果没有找到租户 id 的主机聚合，或者该聚合与放置中的不匹配，则结果将与为请求查找任何合适的主机相同。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 布尔值。 <p>相关选项：</p> <ul style="list-style-type: none"> [scheduler] placement_aggregate_required_for_tenants
max_attempts = 3	整数值	<p>调度尝试的最大数量。</p> <p>这是对给定实例构建/移动操作进行的最多尝试次数。它限制调度程序返回的备用主机数量。当主机列表耗尽时，会引发 MaxRetriesExceeded 异常，并且实例设置为错误状态。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 正整数，其中整数对应于构建或移动实例时可以进行的最多尝试次数。
max_placement_results = 1000	整数值	<p>要请求的最大放置结果数。</p> <p>此设置决定了调度操作期间从放置服务接收的结果的最大限制。它有效地限制可考虑调度与大量候选者匹配的请求的主机数量。</p> <p>值 1（最小值）将有效延迟到放置服务的调度。数值越高，调度程序将在过滤和权重过程中考虑的结果数上限。大型部署可能需要此设置低于可用主机总数，以限制调度程序的内存消耗、网络流量等。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 一个整数，其中整数对应于要返回的放置结果数量。

配置选项 = 默认值	类型	Description
placement_aggregate_required_for_tenants = False	布尔值	<p>需要所有租户的放置聚合关联。</p> <p>当 <code>limit_tenants_to_placement_aggregate=True</code> 时，此设置将控制是否允许没有聚合关联性的租户调度到任何可用节点。如果使用聚合来限制某些租户但并非所有，则这应为 <code>False</code>。如果所有租户都应通过聚合限制，则这应该是 <code>True</code>，以防止它们收到不受限制的调度到任何可用节点。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 布尔值。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [scheduler] placement_aggregate_required_for_tenants
query_placement_for_availability_zone = False	布尔值	<p>使用 <code>placement</code> 确定可用区。</p> <p>此设置会使调度程序使用设置为传入请求所提供的值的元数据键 (availability_zone) 查找主机聚合，并从放置请求请求结果限制为该聚合。</p> <p>必须在放置中镜像匹配的聚合 UUID，才能正常工作。如果没有找到带有 availability_zone 键的主机聚合，或者该聚合与放置中的不匹配，则结果将与找不到任何合适的主机相同。</p> <p>请注意，如果启用此标志，您可以在调度程序中禁用（无效率）<code>AvailabilityZoneFilter</code>。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 布尔值。 <p>相关选项：</p> <ul style="list-style-type: none"> ● [filter_scheduler] enabled_filters
query_placement_for_image_type_support = False	布尔值	<p>使用 <code>placement</code> 来确定对实例的镜像类型的主机支持。</p> <p>此设置会导致调度程序仅为支持请求中使用的镜像的 disk_format 的计算主机请求放置。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 布尔值。

配置选项 = 默认值	类型	Description
query_placement_for_routed_network_aggregates = False	布尔值	启用调度程序过滤从属的计算主机以路由网络段聚合。 详情请查看 https://docs.openstack.org/neutron/latest/admin/config-routed-networks.html 。
Worker = None	整数值	nova-scheduler 服务的 worker 数量。 默认为可用的 CPU 数量。 可能的值： <ul style="list-style-type: none">● 一个整数，其中整数对应于 worker 进程的数量。

9.1.47. serial_console

下表概述了 `/etc/nova/nova.conf` 文件中的 `[serial_console]` 组下可用的选项。

表 9.46. serial_console

配置选项 = 默认值	类型	Description
base_url = ws://127.0.0.1:6083/	URI 值	最终用户用来连接到 nova-serialproxy 服务的 URL。 nova-serialproxy 服务使用此令牌增强 URL 来调用，并建立与正确实例的连接。 相关选项： <ul style="list-style-type: none">● IP 地址必须与 nova-serialproxy 服务侦听的地址相同（请参阅本节中的 serialproxy_host 选项）。● 端口必须与本节的 option serialproxy_port 相同。● 如果您选择使用安全 websocket 连接，则使用 wss:// 启动这个选项，而不是不安全的 ws://。必须为此在 [DEFAULT] 项中设置 cert 和 key 选项。
enabled = False	布尔值	启用串行控制台功能。 要使用这个功能，服务 nova-serialproxy 需要运行。此服务通常在控制器节点上执行。

配置选项 = 默认值	类型	Description
port_range = 10000:20000	字符串值	<p>客户机可用于其后端的 TCP 端口。</p> <p>每个创建的实例都将使用这个范围中的一个端口。如果范围不够大，为新实例提供另一个端口，则不会启动此实例。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 每个通过 regex <code>^\d+:\d+\$</code> 的字符串，例如 10000:20000。确保第一个端口号低于第二个端口号，且范围为 0 到 65535。
proxycient_address = 127.0.0.1	字符串值	<p>代理客户端（如 nova-serialproxy）应连接到的 IP 地址，以获取实例的串行控制台。</p> <p>这通常是 nova-compute 服务的主机的 IP 地址。</p>
serialproxy_host = 0.0.0.0	字符串值	<p>nova-serialproxy 服务用于侦听传入请求的 IP 地址。</p> <p>nova-serialproxy 服务侦听此 IP 地址，以向公开串行控制台的实例侦听传入连接请求。</p> <p>相关选项：</p> <ul style="list-style-type: none"> 确保这与本节的 base_url 选项中定义的 IP 地址定义，或者使用 0.0.0.0 侦听所有地址。
serialproxy_port = 6083	端口值	<p>nova-serialproxy 服务用来侦听传入请求的端口号。</p> <p>nova-serialproxy 服务侦听此端口号，以侦听到向公开串行控制台的实例的请求的传入连接请求。</p> <p>相关选项：</p> <ul style="list-style-type: none"> 确保这是与本节的 base_url 选项 base_url 中定义的端口号。

9.1.48. service_user

下表概述了 `/etc/nova/nova.conf` 文件中的 `[service_user]` 组下可用的选项。

表 9.47. service_user

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围的域名
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称

配置选项 = 默认值	类型	Description
send_service_user_token = False	布尔值	为 True 时，如果向 REST API 发送请求，也会发送服务帐户令牌。 Nova 通常重复利用提供给 nova-api 的用户令牌与 Cinder、Glance 和 Neutron 等其他 REST API 通信。用户令牌在向 Nova 发出请求时可能有效，但令牌可能会在到达其他服务之前过期。为避免出现任何故障，并且要清除它是 Nova 代表用户调用服务，我们包括服务帐户令牌和用户令牌。如果用户的令牌已过期，有效的服务令牌可确保 keystone 中间件仍会接受 REST API 请求。
split-loggers = False	布尔值	将请求记录到多个日志记录器。
system-scope = None	字符串值	系统操作的范围
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名

9.1.49. spice

下表概述了 `/etc/nova/nova.conf` 文件中的 `[spice]` 组下可用的选项。

表 9.48. spice

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
agent_enabled = True	布尔值	<p>在实例上启用 SPICE 客户机代理支持。</p> <p>SPICE 代理与 Spice 协议配合使用，以提供更好的客户机控制台体验。但是，如果未采用 SPICE 代理，仍可使用 Spice 控制台。安装 Spice 代理后，启用了以下功能：</p> <ul style="list-style-type: none"> ● 在客户机和客户端机器间复制和粘贴文本和镜像 ● 当客户端屏幕更改时自动调整分辨率 - 例如，如果您使 Spice 控制台全屏使客户机解析进行调整，则客户机解析将调整为匹配，而不是字母框。 ● 更好的鼠标集成 - 可以捕获和释放鼠标，而无需单击控制台或按键进行释放。也可以提高鼠标移动的性能。
enabled = False	布尔值	<p>启用 SPICE 相关功能。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● 必须明确禁用 VNC 才能访问 SPICE 控制台。在 [vnc] 部分中，将 enabled 选项设置为 False，以禁用 VNC 控制台。

配置选项 = 默认值	类型	Description
html5proxy_base_url = http://127.0.0.1:6082/spice_auto.html	URI 值	<p>SPICE HTML5 控制台代理的位置。</p> <p>最终用户将使用此 URL 连接到 nova-spicehtml5proxy 服务。此服务将请求转发到实例的控制台。</p> <p>要使用 SPICE 控制台，服务 nova-spicehtml5proxy 应该正在运行。此服务通常在控制器节点上启动。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 必须是表单的有效 URL： http://host:port/spice_auto.html，其中 host 是运行 nova-spicehtml5proxy 的节点，端口通常为 6082。考虑不使用默认值，因为它并不适合于任何实际部署。 <p>相关选项：</p> <ul style="list-style-type: none"> ● 这个选项取决于 html5proxy_host 和 html5proxy_port 选项。计算节点返回的访问 URL 必须具有 nova-spicehtml5proxy 服务正在侦听的主机和端口。
html5proxy_host = 0.0.0.0	主机地址值	<p>nova-spicehtml5proxy 服务侦听传入请求的 IP 地址或主机名。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● 这个选项取决于 html5proxy_base_url 选项。nova-spicehtml5proxy 服务必须侦听可从 HTML5 客户端访问的主机。

配置选项 = 默认值	类型	Description
html5proxy_port = 6082	端口值	<p>nova-spicehtml5proxy 服务侦听传入请求的端口。</p> <p>相关选项：</p> <ul style="list-style-type: none"> 这个选项取决于 html5proxy_base_url 选项。nova-spicehtml5proxy 服务必须侦听可从 HTML5 客户端访问的端口。
server_listen = 127.0.0.1	字符串值	<p>在实例上运行的 SPICE 服务器应侦听的地址。</p> <p>通常，nova-spicehtml5proxy 代理客户端在控制器节点上运行，并通过专用网络连接到计算节点上的此地址。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 要侦听的 IP 地址。
server_proxycient_addresses = 127.0.0.1	字符串值	<p>nova-spicehtml5proxy 客户端用于连接实例控制台的地址。</p> <p>通常，nova-spicehtml5proxy 代理客户端在控制器节点上运行，并通过专用网络连接到计算节点上的此地址。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 计算节点上任何有效的 IP 地址。 <p>相关选项：</p> <ul style="list-style-type: none"> 此选项取决于 server_listen 选项。代理客户端必须能够使用此选项的值访问 server_listen 中指定的地址。

9.1.50. upgrade_levels

下表概述了 `/etc/nova/nova.conf` 文件中的 `[upgrade_levels]` 组下可用的选项。

表 9.49. upgrade_levels

配置选项 = 默认值	类型	Description
------------	----	-------------

配置选项 = 默认值	类型	Description
baseapi = None	字符串值	<p>基本 API RPC API 版本大写。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 默认情况下，发送客户端了解的最新版本 ● 代表 <i>N.N</i> 格式的版本的字符串；例如，可能的值可以是 <i>1.12</i> 或 <i>2.0</i>。 ● OpenStack 版本名称，在小写中，如 <i>mitaka</i> 或 <i>liberty</i>。
cert = None	字符串值	<p>Cert RPC API 版本大写。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 默认情况下，发送客户端了解的最新版本 ● 代表 <i>N.N</i> 格式的版本的字符串；例如，可能的值可以是 <i>1.12</i> 或 <i>2.0</i>。 ● OpenStack 版本名称，在小写中，如 <i>mitaka</i> 或 <i>liberty</i>。 <p>弃用自： 18.0.0</p> <p>原因： nova-cert 服务在 16.0.0 (Pike) 中删除，因此不再使用这个选项。</p>
compute = None	字符串值	<p>计算 RPC API 版本大写。</p> <p>默认情况下，我们始终使用客户端了解的最新版本发送消息。</p> <p>如果您运行有旧的计算服务和新的计算服务，应该将其设置为最低部署的版本。这是为了保证所有服务永远不会发送其中一个计算节点无法理解的消息。请注意，我们只支持从 <i>N</i> 版本升级到 <i>N+1</i>。</p> <p>如果要让计算 RPC 模块根据部署中的服务版本自动决定要使用的版本，请将此选项设置为 "auto"。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 默认情况下，发送客户端了解的最新版本 ● <i>自动</i>：根据部署中的服务版本自动决定要使用的版本。 ● 代表 <i>N.N</i> 格式的版本的字符串；例如，可能的值可以是 <i>1.12</i> 或 <i>2.0</i>。 ● OpenStack 版本名称，在小写中，如 <i>mitaka</i> 或 <i>liberty</i>。

配置选项 = 默认值	类型	Description
conductor = None	字符串值	编排器 RPC API 版本上限。 可能的值： <ul style="list-style-type: none"> ● 默认情况下，发送客户端了解的最新版本 ● 代表 <i>N.N</i> 格式的版本的字符串；例如，可能的值可以是 <i>1.12</i> 或 <i>2.0</i>。 ● OpenStack 版本名称，在小写中，如 <i>mitaka</i> 或 <i>liberty</i>。
scheduler = None	字符串值	调度程序 RPC API 版本大写。 可能的值： <ul style="list-style-type: none"> ● 默认情况下，发送客户端了解的最新版本 ● 代表 <i>N.N</i> 格式的版本的字符串；例如，可能的值可以是 <i>1.12</i> 或 <i>2.0</i>。 ● OpenStack 版本名称，在小写中，如 <i>mitaka</i> 或 <i>liberty</i>。

9.1.51. vault

下表概述了 `/etc/nova/nova.conf` 文件中的 `[vault]` 组下可用的选项。

表 9.50. vault

配置选项 = 默认值	类型	Description
aprole_role_id = None	字符串值	Approle role_id 用于使用 vault 进行身份验证
aprole_secret_id = None	字符串值	Approle secret_id 用于使用 vault 进行身份验证
kv_mountpoint = secret	字符串值	要使用的 Vault 中的 KV 存储挂载点，例如： <code>secret</code>
kv_version = 2	整数值	要使用的 Vault 中的 KV 版本，例如： <code>2</code>
root_token_id = None	字符串值	vault 的 root 令牌
ssl_ca_cert_file = None	字符串值	ca 证书文件的绝对路径
use_ssl = False	布尔值	SSL Enabled/Disabled
vault_url = http://127.0.0.1:8200	字符串值	使用此端点连接到 Vault，例如： <code>"http://127.0.0.1:8200"</code>

9.1.52. vendordata_dynamic_auth

下表概述了 `/etc/nova/nova.conf` 文件中的 `[vendordata_dynamic_auth]` 组下可用的选项。

表 9.51. vendordata_dynamic_auth

配置选项 = 默认值	类型	Description
auth-url = None	字符串值	身份验证 URL
auth_section = None	字符串值	从中加载插件特定选项的配置部分
auth_type = None	字符串值	要加载的身份验证类型
cafile = None	字符串值	验证 HTTP 连接时要使用的 PEM 编码证书颁发机构。
certFile = None	字符串值	PEM 编码的客户端证书证书文件
collect-timing = False	布尔值	收集每个 API 调用计时信息。
default-domain-id = None	字符串值	用于 v3 和 v2 参数的可选域 ID。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
default-domain-name = None	字符串值	用于 v3 API 和 v2 参数的可选域名。它将用于 v3 中的用户和项目域，并在 v2 身份验证中忽略。
domain-id = None	字符串值	范围的域 ID
domain-name = None	字符串值	范围为的域名
insecure = False	布尔值	验证 HTTPS 连接。
keyfile = None	字符串值	PEM 编码的客户端证书密钥文件
password = None	字符串值	用户的密码
project-domain-id = None	字符串值	包含项目的域 ID
project-domain-name = None	字符串值	包含项目的域名
project-id = None	字符串值	项目 ID 的范围
project-name = None	字符串值	范围的项目名称
split-loggers = False	布尔值	将请求记录到多个日志记录器。
system-scope = None	字符串值	系统操作的范围

配置选项 = 默认值	类型	Description
tenant-id = None	字符串值	租户 ID
tenant-name = None	字符串值	租户名称
timeout = None	整数值	http 请求的超时值
trust-id = None	字符串值	信任 ID
user-domain-id = None	字符串值	用户的域 ID
user-domain-name = None	字符串值	用户的域名
user-id = None	字符串值	用户 ID
username = None	字符串值	用户名

9.1.53. vmware

下表概述了 `/etc/nova/nova.conf` 文件中的 `[vmware]` 组下可用的选项。

表 9.52. vmware

配置选项 = 默认值	类型	Description
api_retry_count = 10	整数值	连接失败时必须重试 VMware vCenter 服务器 API 的数量，如套接字错误等。
ca_file = None	字符串值	指定用于验证 vCenter 服务器证书时使用的 CA 捆绑包文件。
cache_prefix = None	字符串值	<p>此选项在存储缓存镜像的文件夹中添加一个前缀</p> <p>这不是完整路径 - 只是一个文件夹前缀。只有在计算节点之间共享数据存储缓存时才应使用。</p> <p>注意：这只有在计算节点在同一主机上运行或者具有共享文件系统时才应使用。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表文件夹的缓存前缀的任何字符串
cluster_name = None	字符串值	VMware Cluster ComputeResource 的名称。

配置选项 = 默认值	类型	Description
connection_pool_size = 10	整数值	<p>这个选项设置 http 连接池大小</p> <p>连接池大小是从 nova 到 vSphere 的连接的最大数量。只有在代表连接池已满的警告时，才会增加它，否则默认值应该会增大。</p>
console_delay_seconds = None	整数值	<p>如果受增加的网络延迟影响，请在远程控制台中输入时造成重复字符，则设置这个值。</p>
datastore_regex = None	字符串值	<p>与数据存储名称匹配的正则表达式模式。</p> <p>datastore_regex 设置指定用于 Compute 的数据存储。例如，datastore_regex="nasm" 选择名称以 "nas" 开头的所有数据存储。</p> <div style="display: flex; align-items: flex-start;"> <div style="flex: 1;">  </div> <div style="flex: 2;"> <p>注意</p> <p>如果没有给出 regex，它只是使用最空闲空间选择数据存储。</p> </div> </div> <p>可能的值：</p> <ul style="list-style-type: none"> 任何与数据存储匹配的正则表达式
host_ip = None	主机地址值	<p>用于连接到 VMware vCenter 主机的主机名或 IP 地址。</p>
host_password = None	字符串值	<p>用于连接到 VMware vCenter 主机的密码。</p>
host_port = 443	端口值	<p>连接到 VMware vCenter 主机的端口。</p>
host_username = None	字符串值	<p>用于连接到 VMware vCenter 主机的用户名。</p>
insecure = False	布尔值	<p>如果为 true，则不会验证 vCenter 服务器证书。如果为 false，则使用默认 CA 信任存储进行验证。</p> <p>如果设置了 "ca_file"，则相关选项：如果设置了 "ca_file"，则忽略此选项。</p>
integration_bridge = None	字符串值	<p>只有在使用 NSX-MH Neutron 插件时，才应配置这个选项。这是 ESXi 服务器或主机上的集成网桥名称。这不应该为任何其他 Neutron 插件设置。因此，不会设置默认值。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 代表集成桥接名称的任何有效字符串

配置选项 = 默认值	类型	Description
maximum_objects = 100	整数值	<p>这个选项指定要在单个结果中返回的最大对象数量的限制。</p> <p>当对象计数达到指定限制时，正值将导致操作暂停检索。服务器仍然可以将计数限制为小于配置的值。任何剩余的对象都可以通过额外的请求来检索。</p>
pbm_default_policy = None	字符串值	<p>这个选项指定要使用的默认策略。</p> <p>如果设置了 <code>pbm_enabled</code>，且没有特定请求定义的存储策略，则将使用此策略。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何有效的存储策略，如 VSAN 默认存储策略 <p>相关选项：</p> <ul style="list-style-type: none"> <code>pbm_enabled</code>
pbm_enabled = False	布尔值	<p>这个选项启用或禁用基于实例放置的存储策略。</p> <p>相关选项：</p> <ul style="list-style-type: none"> <code>pbm_default_policy</code>
pbm_wsdl_location = None	字符串值	<p>此选项指定 PBM 服务 WSDL 文件位置 URL。</p> <p>设置这将禁用基于存储策略实例的放置。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何有效的文件路径，如 <code>file:///opt/SDK/spbm/wsdl/pbmService.wsdl</code>
serial_log_dir = /opt/vmware/vspc	字符串值	<p>指定 Virtual Serial Port Concentrator 存储控制台日志文件的目录。它应匹配 VSPC 的 <code>serial_log_dir</code> 配置值。</p>
serial_port_proxy_uri = None	URI 值	<p>标识提供对 <code>serial_port_service_uri</code> 的网络访问的代理服务。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何有效的 URI（方案为 <code>telnet</code> 或 <code>telnet</code>）。 <p>相关选项：如果没有指定 <code>serial_port_service_uri</code>，则忽略此选项。* <code>serial_port_service_uri</code></p>

配置选项 = 默认值	类型	Description
serial_port_service_uri = None	字符串值	<p>标识将发送串行端口流量的远程系统。</p> <p>这个选项添加了一个虚拟串行端口，将控制台输出发送到可配置的服务 URI。在服务 URI 地址中，将收集控制台日志的虚拟串行端口集中器。如果没有设置，则不会将串行端口添加到创建的虚拟机中。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 任何有效的 URI
task_poll_interval = 0.5	浮点值	<p>轮询 VMware VC 服务器上调用的远程任务的时间间隔（以秒为单位）。</p>
use_linked_clone = True	布尔值	<p>这个选项启用/禁用链接克隆的使用。</p> <p>ESX hypervisor 需要 VMDK 文件的副本才能启动虚拟机。计算驱动程序必须通过 HTTP 从 OpenStack 镜像服务下载 VMDK 到 hypervisor 可见的数据存储并将其缓存。需要 VMDK 的后续虚拟机使用缓存版本，且不必从 OpenStack 镜像服务再次复制该文件。</p> <p>如果设置为 false，即使在缓存的 VMDK 中，仍存在从缓存位置复制到共享数据存储中的 hypervisor 文件目录的副本操作。如果设置为 true，则避免上述复制操作，因为它会创建与父虚拟机共享虚拟磁盘的虚拟机副本。</p>
vnc_keymap = en-us	字符串值	<p>VNC 的键映射。</p> <p>键盘映射(keymap)决定 VNC 会话默认应使用的键盘布局。</p> <p>可能的值：</p> <ul style="list-style-type: none"> 此节点上底层管理程序支持的键盘布局。这通常是一个 IETF 语言标签（例如 en-us）。
vnc_port = 5900	端口值	<p>这个选项指定 VNC 启动端口。</p> <p>由 ESX 主机创建的每个虚拟机都有一个选项，可以启用 VNC 客户端进行远程连接。以上选项 vnc_port 可帮助您为 VNC 客户端设置默认启动端口。</p> <p>可能的值：</p> <ul style="list-style-type: none"> Any valid port number within 5900 -(5900 + vnc_port_total) <p>相关选项：应将以下选项设置为启用 VNC 客户端。* vnc.enabled = True * vnc_port_total</p>

配置选项 = 默认值	类型	Description
vnc_port_total = 10000	整数值	VNC 端口的总数。

9.1.54. vnc

下表概述了 `/etc/nova/nova.conf` 文件中的 `[vnc]` 组下可用的选项。

表 9.53. vnc

配置选项 = 默认值	类型	Description
auth_schemes = ['none']	列表值	<p>与计算节点一起使用的身份验证方案。</p> <p>控制代理和计算主机之间的连接允许哪些 RFB 身份验证方案。如果启用了多个方案，则将使用第一个匹配的方案，因此应首先列出最强的方案。</p> <p>相关选项：</p> <ul style="list-style-type: none"> • [VNC]vencrypt_client_key,[vnc]vencrypt_client_cert: 还必须设置
enabled = True	布尔值	<p>启用与 VNC 相关的功能。</p> <p>客户将使用图形设备创建，以支持这一点。然后，客户端（例如 Horizon）可以建立与客户机的 VNC 连接。</p>
novncproxy_base_url = http://127.0.0.1:6080/vnc_auto.html	URI 值	<p>noVNC VNC 控制台代理的公共地址。</p> <p>VNC 代理是一个 OpenStack 组件，它允许计算服务用户通过 VNC 客户端访问其实例。noVNC 通过基于 websocket 的客户端提供 VNC 支持。</p> <p>这个选项设置客户端系统将连接到的公共基本 URL。noVNC 客户端可以使用这个地址连接到 noVNC 实例，并扩展 VNC 会话。</p> <p>如果使用 noVNC >= 1.0.0，您应该使用 vnc_lite.html 而不是 vnc_auto.html。</p> <p>相关选项：</p> <ul style="list-style-type: none"> • novncproxy_host • novncproxy_port

配置选项 = 默认值	类型	Description
novncproxy_host = 0.0.0.0	字符串值	<p>noVNC 控制台代理应绑定到的 IP 地址。</p> <p>VNC 代理是一个 OpenStack 组件，它允许计算服务用户通过 VNC 客户端访问其实例。noVNC 通过基于 websocket 的客户端提供 VNC 支持。</p> <p>此选项设置 noVNC 控制台代理服务应绑定到的专用地址。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● novncproxy_port ● novncproxy_base_url
novncproxy_port = 6080	端口值	<p>noVNC 控制台代理应绑定到的端口。</p> <p>VNC 代理是一个 OpenStack 组件，它允许计算服务用户通过 VNC 客户端访问其实例。noVNC 通过基于 websocket 的客户端提供 VNC 支持。</p> <p>此选项设置 noVNC 控制台代理服务应绑定到的专用端口。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● novncproxy_host ● novncproxy_base_url
server_listen = 127.0.0.1	主机地址值	实例应侦听此节点上的传入 VNC 连接请求的 IP 地址或主机名。
server_proxyclient_addresses = 127.0.0.1	主机地址值	<p>VNC 控制台代理的专用、内部 IP 地址或主机名。</p> <p>VNC 代理是一个 OpenStack 组件，允许计算服务用户通过 VNC 客户端访问其实例。</p> <p>此选项设置代理客户端应连接到的专用地址，如 nova-novncproxy。</p>
vencrypt_ca_certs = None	字符串值	<p>CA 证书 PEM 文件的路径</p> <p>包含一个或多个 x509 证书的 PEM 文件的完全限定路径，用于计算节点 VNC 服务器使用的证书颁发机构。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● VNC.auth_schemes：必须包含 vencrypt

配置选项 = 默认值	类型	Description
vencrypt_client_cert = None	字符串值	<p>客户端密钥文件的路径（用于 x509）</p> <p>包含 x509 证书的 PEM 文件的完全限定路径，在 VNC 身份验证过程中 VNC 代理服务器将其提供给计算节点。</p> <p>Realted 选项：</p> <ul style="list-style-type: none"> ● VNC.auth_schemes：必须包含 vencrypt ● 还必须设置 VNC.vencrypt_client_key：
vencrypt_client_key = None	字符串值	<p>客户端证书 PEM 文件的路径（用于 x509）</p> <p>包含 VNC 代理服务器在 VNC 身份验证过程中向计算节点呈现的私钥的 PEM 文件的完全限定路径。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● VNC.auth_schemes：必须包含 vencrypt ● vnc.vencrypt_client_cert: must also be set

9.1.55. 临时解决方案

下表概述了 `/etc/nova/nova.conf` 文件中的 `[workarounds]` 组下可用的选项。

表 9.54. 临时解决方案

配置选项 = 默认值	类型	描述
disable_compute_service_check_for_ffu = False	布尔值	<p>如果设置了此设置，则旧计算服务的正常安全检查将被视为警告而不是错误。这只是为了促进在计算节点能够更新其服务记录之前启动新控制服务的快速转发升级。在 FFU 中，数据库中的服务记录将早于一个版本，直到计算节点启动为止，但首先需要控制服务在线。</p>

配置选项 = 默认值	类型	描述
disable_fallback_pcpu_query = False	布尔值	<p>在使用固定实例时，禁用对 VCPU 分配的回退请求。</p> <p>从 Train 开始，使用 libvirt virt 驱动程序的计算节点可以报告 PCPU 清单，并将此用于固定实例。调度程序将使用旧的 CPU 固定相关的类别额外规格、hw:cpu_policy 和 hw:cpu_thread_policy、其镜像元数据属性等等的镜像元数据属性以及仿真程序线程固定额外的 spec, hw:emulator_threads_policy 的请求转换为新的放置请求。但是，计算节点需要额外的配置来报告 PCPU 清单，并且升级后可能无法立即存在此配置。为确保在不进行这种额外配置的情况下创建固定实例，调度程序将为旧样式的 VCPU 发出第二个请求，并在需要时回退到这些分配候选版本。这会影响力能，对于在所有主机上设置了新配置或升级的部署来说不需要这样做。通过设置此选项，第二个查找被禁用，调度程序只会请求 PCPU 的分配。</p> <p>弃用自：20.0.0</p> <p>*reason:*None</p>
disable_group_policy_check_upcall = False	布尔值	<p>在 compute 中禁用服务器组策略检查 upcall。</p> <p>为了检测与服务器组关联性策略的竞争条件，计算服务会尝试验证策略没有被调度程序违反。它通过对 API 数据库进行 upcall 来列出要引导的服务器组中的实例，这会违反了我们的 api/cell 隔离目标。最终，这会通过调度程序和放置服务中的正确关联性保证来解决，但直到之后，需要进行此更新检查以确保适当的关联性策略。</p> <p>需要通过这个检查进行 api/cell 隔离的 Operator 应该启用此标志，这样可避免从计算中进行调用。</p> <p>相关选项：</p> <ul style="list-style-type: none"> • [filter_scheduler]/track_instance_changes 还依赖于计算服务到调度程序服务的 upcalls。

配置选项 = 默认值	类型	描述
disable_libvirt_livesnaps hot = False	布尔值	<p>使用 libvirt 驱动程序时禁用实时快照。</p> <p>实时快照允许在不中断客户机的情况下对磁盘进行快照，从而利用与客户机代理的协调来静止文件系统。</p> <p>当使用 libvirt 1.2.2 实时快照时，在负载下间歇性失败（比如与并发 libvirt/qemu 操作相关）。这个配置选项提供了一种禁用实时快照的机制，而选择冷快照，而这会被解决。冷快照会在客户机经历快照进程时导致实例中断。</p> <p>如需更多信息，请参阅错误报告：</p> <p>https://bugs.launchpad.net/nova/+bug/1334398</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：使用 libvirt 时禁用实时快照 ● false：在快照进行快照时始终使用实时快照（只要有新的足够 libvirt 和后端存储支持它） <p>弃用自： 19.0.0</p> <p>reason: 添加了这个选项来临时处理 libvirt 1.2.2 的问题。我们不再支持这个 libvirt 版本，这意味着不再需要这个临时解决方案。它将在以后的发行版本中被删除。</p>

配置选项 = 默认值	类型	描述
disable_native_luksv1 = False	布尔值	<p>将加密的 LUKSv1 Cinder 卷附加到实例时，Libvirt 驱动程序会将加密的磁盘配置为由 QEMU 原生解密。</p> <p>QEMU 使用的 libgcrypt 库中发现了一个性能问题，在这种情况下，服务器会限制 I/O 性能。</p> <p>如需更多信息，请参阅以下 bug 报告：</p> <p>RFE: 硬件加速 AES-XTS 模式 https://bugzilla.redhat.com/show_bug.cgi?id=1762765</p> <p>启用这个临时解决方案选项将导致 Nova 使用传统的基于 os-brick 的 os-brick 加密器来解密 LUKSv1 卷。</p> <p>请注意，在使用不提供主机块设备（如 Ceph）的卷时启用此选项，会导致从实例引导或将卷连接到实例失败。如需了解避免 RBD 执行此操作的方法，请参阅 [workarounds]/rbd_block_device 选项。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● compute_driver (libvirt) ● rbd_block_device (workarounds) <p>弃用自：23.0.0</p> <p>原因： 在 libgcrypt 中提示这个临时解决方案的底层性能回归已被解决，从 1.8.5 开始已解决</p>
disable_rootwrap = False	布尔值	<p>使用 sudo 而不是使用 rootwrap。</p> <p>出于性能的原因，允许回退到 sudo。</p> <p>如需更多信息，请参阅错误报告：</p> <p>https://bugs.launchpad.net/nova/+bug/1415106</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● true：使用 sudo 而不是 rootwrap ● false：照常使用 rootwrap <p>与其他选项间的依赖关系：</p> <ul style="list-style-type: none"> ● 所有影响 <i>rootwrap</i> 的选项都将被忽略。

配置选项 = 默认值	类型	描述
enable_numa_live_migration = False	布尔值	<p>启用使用 NUMA 拓扑的实例实时迁移。</p> <p>只有在完全升级到 Train 的部署中支持使用 libvirt 驱动程序时，使用 NUMA 拓扑的实时迁移实例。在以前的版本中，或在带有滚动升级的混合 Stein/Train 部署中，在使用 libvirt 驱动程序时默认禁用带有 NUMA 拓扑的实例实时迁移。这包括使用 CPU 固定或大页实时迁移实例。目前不会重新计算此类实例的 CPU 固定和巨页信息，如 'bugkamelet89064' 所述。这意味着，如果目标主机上已存在实例，迁移的实例可以放在与这些实例相同的专用内核中，或使用为另一个实例分配的大页。另外，如果主机平台不是同构，可以将实例分配给不存在的内核，或者跨主机 NUMA 节点意外分割。</p> <p>尽管这些已知的问题，在有些情况下可能需要实时迁移。通过启用此选项，了解此问题的 operator 并需要手动工作，可以对这些实例启用实时迁移支持。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● compute_driver：只有 libvirt 驱动程序会受到影响。 <ul style="list-style-type: none"> a. <code>_bug #1289064</code>: https://bugs.launchpad.net/nova/+bug/1289064 <p>弃用自：20.0.0</p> <p><i>*reason*</i> 这个选项被添加来缓解使用 libvirt 驱动程序的 NUMA 拓扑迁移实例时已知的问题。这些问题在 Train 中解决。使用 libvirt 驱动程序的云，并完全升级到 Train 支持 NUMA 感知实时迁移。这个选项将在以后的发行版本中被删除。</p>
enable_qemu_monitor_announce_self = False	布尔值	<p>如果 libvirt 驱动程序设为 True，则 libvirt 驱动程序将尝试作为向 QEMU 监视器发送声明命令的最佳努力，以便它生成 RARP 帧以更新目标后实时迁移阶段的网络交换机。</p> <p>请注意，这会导致 libvirt 认为域被 libvirt 污点。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>:oslo.config:option:DEFAULT.compute_driver</code> (libvirt)

配置选项 = 默认值	类型	描述
ensure_libvirt_rbd_instance_dir_cleanup = False	布尔值	<p>在使用 rbd 时，确保实例目录在清理过程中移除。</p> <p>启用此临时解决方案后，请确保使用 [libvirt]/images_type=rbd 的主机上始终删除实例目录。这可避免以下错误，并显示撤离并恢复调整导致主机上剩余的实例目录的大小：</p> <p>https://bugs.launchpad.net/nova/+bug/1414895</p> <p>https://bugs.launchpad.net/nova/+bug/1761062</p> <p>然后，如果实例试图返回到主机，则这两个错误都可能会导致 DestinationDiskExists 错误。</p> <p>a. 警告：Operator 需要确保由 [DEFAULT]/instances_path 指定的实例目录本身不会在计算之间共享，然后再启用这个临时解决方案，否则 console.log、内核、ramdisk 以及正在运行的实例使用的任何其他文件将会丢失。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● compute_driver (libvirt) ● [libvirt]/images_type (rbd) ● instances_path

配置选项 = 默认值	类型	描述
handle_virt_lifecycle_events = True	布尔值	<p>启用处理计算驱动程序发送的事件。</p> <p>许多计算驱动程序发出生命周期事件，这些事件是实例启动或停止时发生的事件。如果实例因为 API 操作而经历了任务状态的变化，则忽略事件。</p> <p>这是一个高级功能，它允许虚拟机监控程序向计算服务发出意外状态更改的信号，并且实例可以自动关闭。不幸的是，这可能会在某些条件下争用，例如在重启操作中，或者在主机重新引导时或主机重新引导时（计划或因中断而出现）。如果此类竞争很常见，建议禁用此功能。</p> <p>当禁用这个功能且 <code>sync_power_state_interval</code> 设置为负值时，应小心。在这种情况下，虚拟机监控程序和 Nova 数据库之间不同步的任何实例都必须手动同步。</p> <p>如需更多信息，请参阅错误报告： https://bugs.launchpad.net/bugs/1444630</p> <p>与其他选项间的依赖关系：</p> <ul style="list-style-type: none"> ● 如果 sync_power_state_interval 为负状态，且此功能被禁用，那么管理程序和 Nova 数据库之间的不同步的实例必须手动同步。
libvirt_disable_apic = False	布尔值	<p>在一些内核初始化 guest apic 时，可能会导致内核挂起，导致客户机不可用。这会导致内核错误。在大多数情况下，正确的修复它会将客户机镜像内核更新至一个已修补的客户机镜像内核，但在某些情况下无法进行修复。此临时解决方案允许模拟每个主机禁用 apic，但不建议在 CI 或开发人员云之外使用。</p>
never_download_image_if_on_rbd = False	布尔值	<p>从由 ceph 支持的计算节点上的镜像引导时，如果镜像还没有驻留在 ceph 集群中（如果 glance 也使用相同的集群），nova 将从 glance 下载镜像并将其上传到 ceph 本身。如果使用多个 ceph 集群，这可能会导致 nova 在本地 ceph 部署中意外地复制镜像。</p> <p>如需更多信息，请参阅错误报告： https://bugs.launchpad.net/nova/+bug/1858877</p> <p>启用此选项将导致 nova 在需要从 glance 下载镜像并将其上传到 ceph 本身时 拒绝 引导实例。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● compute_driver (libvirt) ● [libvirt]/images_type (rbd)

配置选项 = 默认值	类型	描述
rdp_volume_local_attach = False	布尔值	<p>将 RBD Cinder 卷作为主机块设备附加到计算。</p> <p>启用此选项后，指示 os-brick 将计算主机上本地的 RBD 卷连接为块设备，而不是通过 QEMU 进行原生连接。</p> <p>这个临时解决方案目前不支持扩展附加的卷。</p> <p>这可与 <code>disable_native_luksv1</code> 临时解决方案配置选项一起使用，以避免在 <code>libgcrypt</code> 库中找到的最近发现的性能问题。</p> <p>这个临时解决方案是临时的，当所有受影响的发行版本都能够更新其 <code>libgcrypt</code> 库后，会在 W 版本中删除这个临时解决方案。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● compute_driver (libvirt) ● disable_qemu_native_luksv1 (workarounds) <p>弃用自：23.0.0</p> <p>原因：在 <code>libgcrypt</code> 中提示这个临时解决方案的底层性能回归已被解决，从 1.8.5 开始已解决</p>
reserve_disk_resource_for_image_cache = False	布尔值	<p>如果设置为 True，则 libvirt 驱动程序将为镜像缓存中存储的镜像保留 DISK_GB 资源。如果 <code>:oslo.config:option:DEFAULT.instances_path</code> 位于与镜像缓存目录不同的磁盘分区上，则驱动程序不会为缓存保留资源。</p> <p>此类磁盘保留由每个 <code>:oslo.config:option:update_resources_interval</code> 秒运行的资源跟踪器中的一个定期任务完成。因此，当镜像被缓存时，不会立即更新保留。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● <code>:oslo.config:option:DEFAULT.instances_path</code> ● <code>:oslo.config:option:image_cache.subdirectory_name</code> ● <code>:oslo.config:option:update_resources_interval</code>
skip_cpu_compare_at_startup = False	布尔值	<p>这将在 Compute 服务启动时跳过 CPU 比较调用，并允许 libvirt 处理它。</p>

配置选项 = 默认值	类型	描述
skip_cpu_compare_on_dest = False	布尔值	启用后，它将跳过目标主机上的 CPU 比较。当使用 QEMU >= 2.9 和 libvirt >= 4.4.0 时，libvirt 将对实时迁移过程中检查目标主机上的 CPU 兼容性进行正确的操作。
skip_hypervisor_version_check_on_lm = False	布尔值	启用后，它将在实时迁移过程中跳过虚拟机监控程序的版本检查。

配置选项 = 默认值	类型	描述
<p>wait_for_vif_plugged_event_during_hard_reboot = []</p>	<p>列表值</p>	<p>libvirt virt 驱动程序通过停止重新引导实例的每个 vif 来实施电源和硬重启，然后再次插入它们。默认情况下，nova 不会等待来自 neutron 的 network-vif-plugged 事件，然后让实例运行。这可能会导致实例通过 DHCP 请求 IP，然后 neutron 后端有机会在 vif 插件后设置网络后端。</p> <p>此标志定义在硬重启过程中需要 network-vif-plugged 事件。可能的值有 neutron port vnic 类型：</p> <ul style="list-style-type: none"> ● normal ● direct ● macvtap ● baremetal ● direct-physical ● virtio-forwarder ● smart-nic ● vdpa ● accelerator-direct ● accelerator-direct-physical <p>向此配置中添加 vnic_type 会导致 Nova 等待每个实例的 vifs 的 vifs 在取消暂停实例之前具有特定 vnic_type 的 vifs 等待网络插入事件，类似于新实例创建的工作方式。</p> <p>请注意，并非所有 neutron 网络后端都会为某些 vnic_type 发送插件时间事件，因此此配置默认为空。</p> <p>ml2/ovs 和 networking-odl 后端已知为带有 normal vnic_type 的端口发送插件时间事件，因此如果您在计算主机上只使用这些后端，则可以安全地将此配置 正常 添加。</p> <p>在带有 direct vnic_type 的端口期间，neutron in-tree SRIOV 后端不会可靠地发送 network-vif-plugged 事件，且永远不会在插件期间发送具有 直接物理 vnic_type 的端口的事件。对于其他 vnic_type 和 backend 对，请查阅后端开发人员。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● :oslo.config:option:DEFAULT.vif_plugging_timeout

9.1.56. wsgi

下表概述了 `/etc/nova/nova.conf` 文件中的 `[wsgi]` 组下可用的选项。

表 9.55. wsgi

配置选项 = 默认值	类型	Description
api_paste_config = api-paste.ini	字符串值	此选项代表 nova-api 的 paste.deploy 配置的文件名。 可能的值： <ul style="list-style-type: none"> ● 代表 paste.deploy 配置的文件名的字符串。
client_socket_timeout = 900	整数值	这个选项指定客户端连接的套接字操作的超时时间。如果传入的连接闲置了这个秒数，它将被关闭。它表示套接字连接中的单独读/写的超时。等待ever 设置为 0。
default_pool_size = 1000	整数值	这个选项指定 wsgi 使用的 greenthreads 池的大小。可以使用此选项限制并发连接数。
keep_alive = True	布尔值	这个选项允许使用相同的 TCP 连接来发送和接收多个 HTTP 请求/响应，而不是为每个单个请求/响应打开一个新的请求/响应。HTTP keep-alive 表示 HTTP 连接重复使用。 可能的值： <ul style="list-style-type: none"> ● true : 重复使用 HTTP 连接。 ● false : 明确关闭客户端套接字连接。 相关选项： <ul style="list-style-type: none"> ● tcp_keepidle
max_header_line = 16384	整数值	此选项指定要接受的消息标头的最大行大小。在使用带有大型服务目录的 Keystone v3 API 生成时，可能需要增加 max_header_line 的最大行大小。 由于 TCP 是一种基于流的协议，因此为了重复使用连接，因此 HTTP 必须有一种方式来指示之前响应的结束和下一个开始。因此，在 keep_alive 情况下，所有消息都必须具有自我定义的消息长度。

配置选项 = 默认值	类型	Description
<p>secure_proxy_ssl_header = None</p>	<p>字符串值</p>	<p>这个选项指定用于确定原始请求的协议方案的 HTTP 标头，即使它是由 SSL 终止代理删除的。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● none（默认）- 请求方案不受任何 HTTP 标头的影响 ● 有效的 HTTP 标头，如 HTTP_X_FORWARDED_PROTO <div data-bbox="817 586 1428 875" style="background-color: #fff9c4; padding: 10px; border: 1px solid #ccc;"> <p> 警告</p> <p>除非您知道要做什么，否则请不要设置此项。</p> </div> <p>在设置此选项前，请确保以下所有都为 true（假设上例中的值）：</p> <ul style="list-style-type: none"> ● 您的 API 位于代理后面。 ● 您的代理从所有传入的请求中剥离 X-Forwarded-Proto 标头。换句话说，如果最终用户在其请求中包含该标头，代理会丢弃它。 ● 您的代理设置 X-Forwarded-Proto 标头并将其发送到 API，但只适用于最初通过 HTTPS 附带的请求。 <p>如果其中任何一个不是 true，则把此设置设置为 None。</p>
<p>ssl_ca_file = None</p>	<p>字符串值</p>	<p>此选项允许设置到应该用于验证连接客户端的 CA 证书文件的路径。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表 CA 证书文件的路径的字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> ● enabled_ssl_apis

配置选项 = 默认值	类型	Description
ssl_cert_file = None	字符串值	<p>此选项允许设置到 API 服务器的 SSL 证书的路径。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表到 SSL 证书的路径的字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> ● enabled_ssl_api
ssl_key_file = None	字符串值	<p>这个选项指定在 SSL 生效时存储 API 服务器的 SSL 私钥的文件的文件的路径。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● 代表到 SSL 私钥的路径的字符串。 <p>相关选项：</p> <ul style="list-style-type: none"> ● enabled_ssl_api
tcp_keepidle = 600	整数值	<p>这个选项为每个服务器套接字设置 TCP_KEEPIDLE 的值（以秒为单位）。它指定保持连接处于活跃状态的时间持续时间。TCP 为请求保持连接活跃的应用程序生成 KEEPALIVE 传输。在 OS X 上不支持。</p> <p>相关选项：</p> <ul style="list-style-type: none"> ● keep_alive
wsgi_log_format = %(client_ip) s "%(request_line) s" status: %(status_code) s len: %(body_length) s time: %(wall_seconds).7f	字符串值	<p>它代表了一个 python 格式字符串，用作模板来生成日志行。以下值可以格式化为：client_ip, date_time, request_line, status_code, body_length, wall_seconds。</p> <p>此选项用于在 eventlet 下运行 nova-api 时构建自定义请求日志。如果在 uwsgi 或 apache 下使用，这个选项无效。</p> <p>可能的值：</p> <ul style="list-style-type: none"> ● <code>%(client_ip) s "%(request_line) s" status: %(status_code) s 'len: %(body_length) s time: %(wall_seconds).7f (default)</code> ● 由特定值组成的任何格式化字符串。 <p>弃用自： 16.0.0</p> <p>reason： 此选项仅在 eventlet 下运行 nova-api 时起作用，并对非常特定于事件的信息进行编码。从 Pike 开始，运行 nova-api 的首选模型位于 uwsgi 或 apache mod_wsgi 下。</p>

9.1.57. zvm

下表概述了 `/etc/nova/nova.conf` 文件中的 `[zvm]` 组下可用的选项。

表 9.56. zvm

配置选项 = 默认值	类型	Description
<code>ca_file = None</code>	字符串值	在启用了 TLS 的 httpd 服务器中验证的 CA 证书文件字符串，它必须是要使用的 CA 捆绑包的路径。
<code>cloud_connector_url = None</code>	URI 值	用于与 z/VM 云连接器通信的 URL。
<code>image_tmp_path = \$state_path/images</code>	字符串值	<p>镜像要存储的路径（快照、部署等）。</p> <p>用于部署和通过快照捕获的镜像需要存储在计算主机的本地磁盘中。此配置标识目录位置。</p> <p>可能的值：运行计算服务的主机上的文件系统路径。</p>
<code>reachable_timeout = 300</code>	整数值	<p>等待实例启动超时（秒）。</p> <p>z/VM 驱动程序依赖于实例和云连接器之间的通信。创建实例后，必须有足够的时间等待所有网络信息写入用户目录中。该驱动程序会将网络状态重新检查到带有超时值的实例，如果设置网络失败，它将通知用户启动实例失败，并将实例置于 ERROR 状态。然后会删除底层 z/VM 客户机。</p> <p>可能的值：任何正整数。建议至少为 300 秒(5 分钟)，但它会根据实例和系统负载而有所不同。值 0 用于 debug。在这种情况下，当实例被标记为 ERROR 状态时，底层 z/VM 客户机不会被删除。</p>